Oracle® Fusion Middleware

Enterprise Deployment Guide for Oracle Identity and Access Management

12c (12.2.1.4.0)

F20762-08

January 2025

Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Identity and Access Management, 12c (12.2.1.4.0)

F20762-08

Primary Author: Oracle Corporation

Contents

Preface

Part I Understanding an Enterprise Deployment

1 Enterprise Deployment Overview

2 About a Typical Enterprise Deployment

3 About the IAM Enterprise Deployment

4 About a Multi-Data Center Deployment

Part II Preparing for an Enterprise Deployment

5 Using the Enterprise Deployment Workbook

6 Procuring Resources for an Enterprise Deployment

7 Preparing the Load Balancer and Firewalls for an Enterprise Deployment

8 Preparing the File System for an Enterprise Deployment

9 Preparing the Operating System for an Oracle Identity and Access Management Deployment

10 Preparing the Host Computers for an Enterprise Deployment

11 Preparing the Database for an Enterprise Deployment

Part III Configuring the Enterprise Deployment

12 Configuring Oracle LDAP for an Enterprise Deployment

17 Configuring Oracle Identity Governance

18 Configuring Multi-Data Center

Part IV Common Configuration and Management Procedures for an Enterprise Deployment

19 Common Configuration and Management Tasks for an Enterprise Deployment

20 Using Whole Server Migration and Service Migration in an Enterprise Deployment

21 Scaling Procedures for an Enterprise Deployment

23 Sanity Checks

24 Troubleshooting

Preface

This guide explains how to install, configure, and manage a highly available Oracle Fusion Middleware enterprise deployment.

- Audience
- Documentation Accessibility
- Conventions
- Diversity and Inclusion

Audience

In general, this document is intended for administrators of Oracle Fusion Middleware, who are assigned the task of installing and configuring Oracle Fusion Middleware software for production deployments.

Specific tasks can also be assigned to specialized administrators, such as database administrators (DBAs) and network administrators, where applicable.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `https://www.oracle.com/corporate/accessibility/`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `https://support.oracle.com/portal/` or visit `Oracle Accessibility Learning and Support` if you are hearing impaired.

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.

Note:

This guide focuses on the implementation of the enterprise deployment reference topology on Oracle Linux systems.

The topology can be implemented on any certified, supported operating system, but the examples in this guide typically show the commands and configuration steps as they should be performed using the bash shell on Oracle Linux.

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

Understanding an Enterprise Deployment

It is important to understand the concept and general characteristics of a typical enterprise deployment, before you configure the Oracle Identity and Access Management enterprise deployment topology.

This part of the Enterprise Deployment Guide contains the following topics.

- Enterprise Deployment Overview
  The Enterprise Deployment Guide provides detailed, validated instructions that help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.
- About a Typical Enterprise Deployment
  The illustration of a typical enterprise deployment topology helps you understand the components of the topology. The topology consists of a web tier, application tier, and data tier.
- About the IAM Enterprise Deployment
- About a Multi-Data Center Deployment
  This chapter describes the multi-data center active-passive or active-passive disaster protection and the multi-data center active-active or active-active disaster protection.

1 Enterprise Deployment Overview

The Enterprise Deployment Guide provides detailed, validated instructions that help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.

This chapter introduces the concept of an Oracle Fusion Middleware enterprise deployment. It also provides information on when to use the Enterprise Deployment guide.

- About the Enterprise Deployment Guide
  An Enterprise Deployment Guide provides a comprehensive, scalable example for installing, configuring, and maintaining a secure, highly available, production-quality deployment of selected Oracle Fusion Middleware products. The resulting environment is known as an **enterprise deployment topology**.
- When to Use the Enterprise Deployment Guide
  This guide describes one of the three primary installation and configuration options for Oracle Fusion Middleware. Use this guide to help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.

About the Enterprise Deployment Guide

An Enterprise Deployment Guide provides a comprehensive, scalable example for installing, configuring, and maintaining a secure, highly available, production-quality deployment of selected Oracle Fusion Middleware products. The resulting environment is known as an **enterprise deployment topology**.

For example, the enterprise deployment topology introduces key concepts and best practices that you can use to implement a similar Oracle Fusion Middleware environment for your organization.

Each Enterprise Deployment Guide provides detailed, validated instructions for implementing the reference topology. Along the way, the guide also offers links to supporting documentation that explains concepts, reference material, and additional options for an Oracle Fusion Middleware enterprise deployment.

Note that the enterprise deployment topologies described in the enterprise deployment guides cannot meet the exact requirements of all Oracle customers. In some cases, you can consider alternatives to specific procedures in this guide, depending on whether the variations to the topology are documented and supported by Oracle.

Oracle recommends customers use the Enterprise Deployment Guides as a first option for deployment. If variations are required, then those variations should be verified by reviewing the related Oracle documentation or by working with Oracle Support.

When to Use the Enterprise Deployment Guide

This guide describes one of the three primary installation and configuration options for Oracle Fusion Middleware. Use this guide to help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.

Alternatively, you can use the other primary installation and configuration options.

Use the instructions in one of the product-specific installation guides to install and configure a **standard installation topology** for a selected set of Oracle Fusion Middleware products.

A standard installation topology can be installed on a single host for evaluation purposes, but it can also serve as a starting point for scaling out to a more complex production environment.

For Oracle Identity and Access Management, see:

For Oracle Identity and Access Management, see Installing and Configuring Oracle Identity and Access Management

Review *Planning an Installation of Oracle Fusion Middleware*, which provides additional information to help you prepare for any Oracle Fusion Middleware installation.

2

About a Typical Enterprise Deployment

The illustration of a typical enterprise deployment topology helps you understand the components of the topology. The topology consists of a web tier, application tier, and data tier.

This chapter includes the following topics:

- Diagram of a Typical Enterprise Deployment
  This diagram shows all the components of a typical enterprise deployment, including the Web tier, Application tier, and Data tier. All enterprise deployments are based on these basic principles.
- About the Typical Enterprise Deployment Topology Diagram
  A typical enterprise deployment topology consists of a Hardware Load Balancer (LBR), web tier, an application tier, and data tier. This section provides detailed information on these components.

Diagram of a Typical Enterprise Deployment

This diagram shows all the components of a typical enterprise deployment, including the Web tier, Application tier, and Data tier. All enterprise deployments are based on these basic principles.

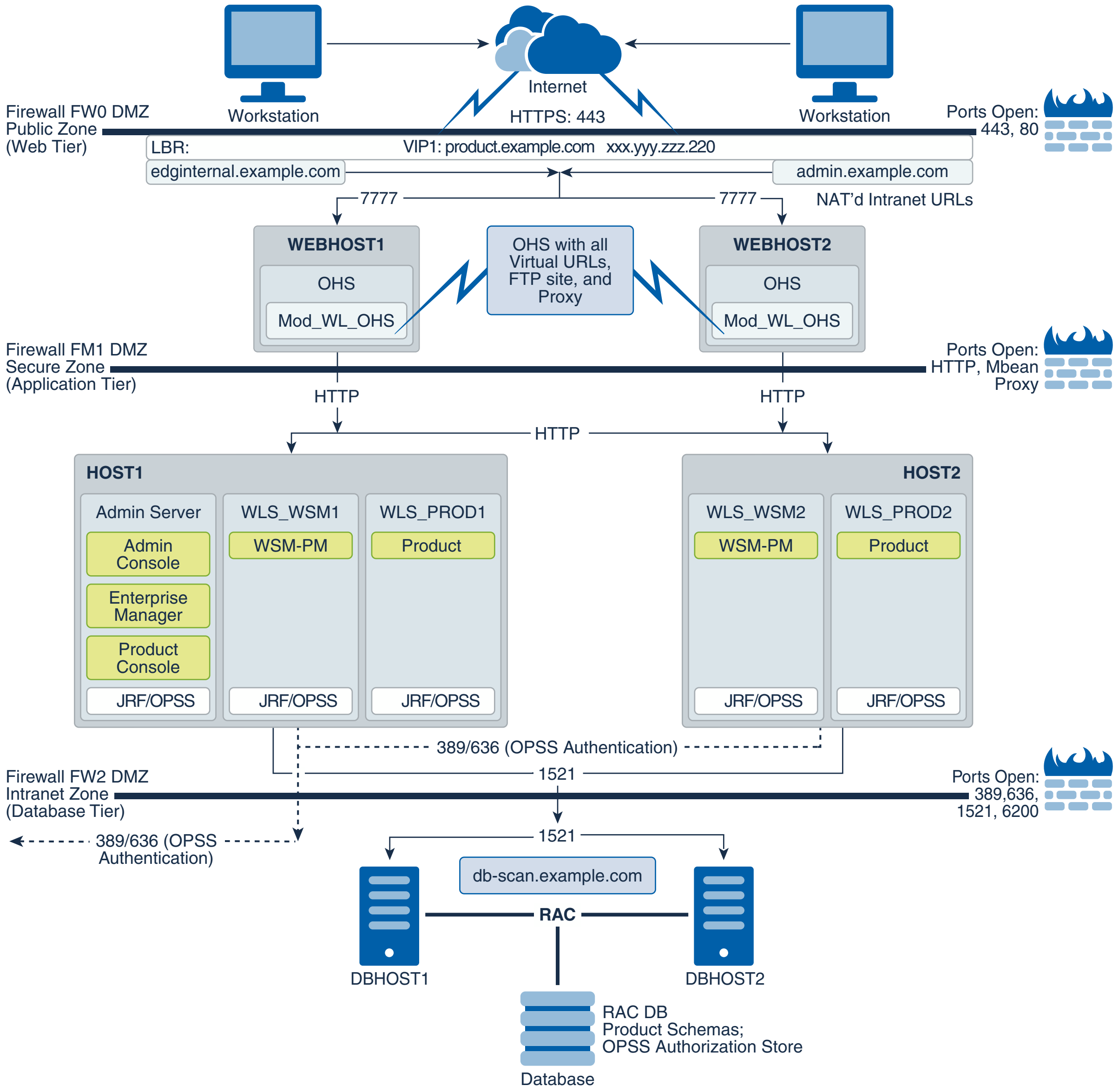
All Oracle Fusion Middleware enterprise deployments are designed to demonstrate the best practices for installing and configuring an Oracle Fusion Middleware production environment.

A best practices approach starts with the basic concept of a multi-tiered deployment and standard communications between the different software tiers.

Figure 2-1 shows a typical enterprise deployment, including the Web tier, Application tier, and Data tier. All enterprise deployments are based on these basic principles.

For a description of each tier and the standard protocols used for communications within a typical Oracle Fusion Middleware enterprise deployment, see About the typical Enterprise Deployment Topology Diagram.

Figure 2-1 Typical Enterprise Deployment Topology Diagram



About the Typical Enterprise Deployment Topology Diagram

A typical enterprise deployment topology consists of a Hardware Load Balancer (LBR), web tier, an application tier, and data tier. This section provides detailed information on these components.

- Understanding the Firewalls and Zones of a Typical Enterprise Deployment
- About the Elements of a Typical Enterprise Deployment Topology
- Receiving Requests Through Hardware Load Balancer
- About Web Tier
- About the Application Tier

- About the Data Tier

Understanding the Firewalls and Zones of a Typical Enterprise Deployment

The topology is divided into several security zones, which are separated by firewalls:

- The web tier (or DMZ), which is used for the hardware load balancer and Web servers (in this case, Oracle HTTP Server instances) that receive the initial requests from users. This zone is accessible only through a single virtual server name that is defined on the load balancer.
- The application tier, which is where the business and application logic resides.
- The data tier, which is not accessible from the Internet and reserved in this topology for the highly available database instances.

The firewalls are configured to allow data to be transferred only through specific communication ports. Those ports (or in some cases, the protocols that need open ports in the firewall) are shown on each firewall line in the diagram.

For example:

- On the firewall protecting the web tier, only the HTTP ports are open: 443 for HTTPS and 80 for HTTP.
- On the firewall protecting an application tier, HTTP ports, and MBean proxy port are open.

  Applications that require external HTTP access can use the Oracle HTTP Server instances as a proxy. Note that this port for outbound communications only and the proxy capabilities on the Oracle HTTP Server must be enabled.
- On the firewall protecting the data tier, the database listener port (typically, 1521) must be open.

  The LDAP ports (typically, 389 and 636) are also required to be open for communication between the authorization provider and the LDAP-based identity store.

  The ONS port (typically, 6200) is also required so that the application tier can receive notifications about workload and events in the Oracle RAC Database. These events are used by the Oracle WebLogic Server connection pools to adjust quickly (creating or destroying connections), depending on the availability and workload on the Oracle RAC database instances.

For a complete list of the ports that you must open for a specific Oracle Fusion Middleware enterprise deployment topology, see the chapter that describes the topology that you want to implement, or refer to the *Enterprise Deployment Workbook* for the topology that you want to implement. See Using the Enterprise Deployment Workbook.

About the Elements of a Typical Enterprise Deployment Topology

The enterprise deployment topology consists of the following high-level elements:

- A hardware load balancer that routes requests from the Internet to the web servers in the web tier. It also routes requests from internal clients or other components that perform internal invocations within the corporate network.
- A web tier, consisting of a hardware load balancer and two or more physical computers that host the web server instances (for high availability).

  The web server instances are configured to authenticate users (through an external identity store and a single sign-on server) and then route the HTTP requests to the Oracle Fusion Middleware products and components that are running in the Application tier.

The web server instances also host static web content that does not require the application logic to be delivered. Placing such content in the web tier reduces the overhead on the application servers and eliminates unnecessary network activity.

- An application tier, consisting of two or more physical computers that are hosting a cluster of Oracle WebLogic Managed Servers, and the Administration Server for the domain. The Managed Servers are configured to run the various Oracle Fusion Middleware products, such as Oracle SOA Suite, Oracle Service Bus, Oracle WebCenter Content, and Oracle WebCenter Portal, depending on your choice of products in the enterprise deployment.
- A data tier, consisting of two or more physical hosts that are hosting an Oracle RAC Database.

Receiving Requests Through Hardware Load Balancer

The following topics describe the hardware load balancer and its role in an enterprise deployment.

- Purpose of the Hardware Load Balancer (LBR)
- Summary of the Typical Load Balancer Virtual Server Names
- HTTPS Versus HTTP Requests to the External Virtual Server Name

Purpose of the Hardware Load Balancer (LBR)

There are two types of load balancers, Local Load Balancers and Global Load Balancers. Load balancers can either be hardware devices such as Big IP, Cisco, Brocade, and so on—or they can be software applications such as Oracle Traffic Director. Most load balancer appliances can be configured for both local and global load balancers.

Load balancers should always be deployed in pairs to ensure that no single load balancer is a single point of failure. Most load balancers do this in an active-passive way. You should consult your load balancer documentation on how best to achieve this.

> **Note:**
>
> Oracle does not certify against specific load balancers. The configuration information of load balancers given in the Enterprise Deployment guide are for guidance only and you should consult with your load balancer vendor about the best practices that are associated with the configuration of the device that you are using.

A local load balancer is used to distribute traffic within a site. It can distribute both HTTP and TCP traffic and the requirements of your deployment dictates which options you should use. Local load balancers often provide acceleration for SSL encryption and decryption as well as the ability to terminate or `off-load` SSL requests. SSL termination at the load balancer provides a significant performance gain to applications, ensuring that traffic to and from a site remains encrypted without the overhead of on the fly software encryption inside the deployment itself. Enterprise Deployment guide environments always utilize a local load balancer.

A global load balancer is used when you have multiple sites that need to function as the same logical environment. Its purpose is to distribute requests between the sites based on a pre-determined set of rules. Global load balancers are typically used in Disaster Recovery (DR) deployments or Active/Active Multi-Data Center (MDC) deployments.

The following topics describe the types of requests that are handled by the hardware load balancer in an Enterprise Deployment:

- HTTP Requests From the Internet to the Web Server Instances in the Web Tier
- Specific Internal-Only Communications Between the Components of the Application Tier
- Load Balancer Considerations for Disaster Recovery and Multi-Data Center Topologies

HTTP Requests From the Internet to the Web Server Instances in the Web Tier

The hardware load balancer balances the load on the web tier by receiving requests to a single virtual host name and then routing each request to one of the web server instances, based on a load balancing algorithm. In this way, the load balancer ensures that no one web server is overloaded with HTTP requests.

For more information about the purpose of specific virtual host names on the hardware load balancer, see Summary of the Typical Load Balancer Virtual Server Names.

Note that in the reference topology, only HTTP requests are routed from the hardware load balancer to the web tier. Secure Socket Layer (SSL) requests are terminated at the load balancer and only HTTP requests are forwarded to the Oracle HTTP Server instances. This guide does not provide instructions for SSL configuration between the load balancer and the Oracle HTTP Server instances or between the web tier and the application tier.

The load balancer provides high availability by ensuring that if one web server goes down, requests are routed to the remaining web servers that are up and running.

Further, in a typical highly available configuration, the hardware load balancers are configured such that a hot standby device is ready to resume service in case a failure occurs in the main load balancing appliance. This is important because for many types of services and systems, the hardware load balancer becomes the unique point of access to make invocations and, as a result, becomes a single point of failure (SPOF) for the whole system if it is not protected.

Specific Internal-Only Communications Between the Components of the Application Tier

In addition, the hardware load balancer routes specific communications between the Oracle Fusion Middleware components and applications on the application tier. The internal-only requests are also routed through the load balancer by using a unique virtual host name.

Load Balancer Considerations for Disaster Recovery and Multi-Data Center Topologies

In addition to the load-balancing features for local site traffic as described in the previous topics, many LBR also include features for configuring global load-balancing across multiple sites in DR or active/active MDC topologies.

A global load balancer configuration uses conditional DNS to direct traffic to local load balancers at different sites. A global load balancer for Oracle Fusion Middleware is typically configured for DR or MDC topologies:

- Active/Passive DR: Always send requests to site 1 unless site 1 in unavailable in which case send traffic to site 2.
- Active/Active MDC: Always send requests to both site 1 and site 2, often based on the geographic location of the source request in relation to the physical geographical location of the sites. Active/Active deployments are available only to those applications which support it.

For example:

```
Application entry point:  app.example.com

Site 1 - Local Load Balancer Virtual Host:  site1app.example.com
Site 2 - Local Load Balancer Virtual Host:  site2app.example.com
```

When a request for `app.example.com` is received, the global load balancer would:

- If the topology is active/passive DR:

  Change the IP address of `app.example.com` in DNS to resolve as the IP address of the local load balancer Virtual Host for the active site. For example: `site1app.example.com` (assuming that is the active site).
- If the topology is active/active MDC:

  Change the IP address of `app.example.com` in DNS to resolve as either the IP address of `site1app.example.com` or `site2app.example.com` depending on which site is nearest to the client making the request.

For information on Disaster Recovery, see *Disaster Recovery Guide*.

For more information on Multi-Data Center topologies for various Fusion Middleware products, see the MAA Best Practices for Fusion Middleware page on the Oracle Technology Network website.

Summary of the Typical Load Balancer Virtual Server Names

In order to balance the load on servers and to provide high availability, the hardware load balancer is configured to recognize a set of virtual server names. By using the naming convention in Figure 2-1, the following virtual server names are recognized by the hardware load balancer in this topology:

- `product.example.com`: This virtual server name is used for all incoming traffic.

  Users enter this URL to access the Oracle Fusion Middleware product that you have deployed and the custom applications that are available on this server. The load balancer then routes these requests (by using a load balancing algorithm) to one of the servers in the web tier. In this way, the single virtual server name can be used to route traffic to multiple servers for load balancing and high availability of the web servers instances.
- `productinternal.example.com`: This virtual server name is for internal communications only.

  The load balancer uses its **Network Address Translation (NAT)** capabilities to route any internal communication from the application tier components that are directed to this URL. This URL is not exposed to external customers or users on the Internet. Each product has specific uses for the internal URL, so in the deployment instructions, the virtual server name is prefixed with the product name.
- `admin.example.com`: This virtual server name is for administrators who need to access the Oracle Enterprise Manager Fusion Middleware Control and Oracle WebLogic Server Administration Console interfaces.

  This URL is known only to internal administrators. It also uses the NAT capabilities of the load balancer to route administrators to the active Administration Server in the domain.

For a complete set of virtual server names that you must define for your topology, see the chapter that describes the product-specific topology.

HTTPS Versus HTTP Requests to the External Virtual Server Name

Note that when you configure the hardware load balancer, a best practice is to assign the main external URL (for example, `http://myapplication.example.com`) to port 80 and port 443.

Any request on port 80 (non-SSL protocol) should be redirected to port 443 (SSL protocol). Exceptions to this rule include requests from public WSDLs. See Configuring Virtual Hosts on the Hardware Load Balancer.

About Web Tier

The web tier of the reference topology consists of web servers that receive requests from the load balancer. In a typical enterprise deployment, at least two Oracle HTTP Server instances are configured in the web tier. The following topics provide more detail.

- Benefits of Using a Web Tier to Route Requests
- Alternatives to Using a Web Tier
- Configuration of Oracle HTTP Server in the Web Tier
- About Mod_WL_OHS

Benefits of Using a Web Tier to Route Requests

A web tier with Oracle HTTP Server is not a requirement for many of the Oracle Fusion Middleware products. You can route traffic directly from the hardware load balancer to the WLS servers in the Application Tier. However, a web tier provides several advantages, which is why it is recommended as part of the reference topology.

- The web tier provides faster fail-over in the event of a WebLogic Server instance failure. The plug-in actively learns about the failed WebLogic Server instance by using the information supplied by its peers. It avoids the failed server until the peers notify the plug-in that it is available. Load balancers are typically more limited and their monitors cause higher overhead.
- The web tier provides DMZ public zone, which is a common requirement in security audits. If a load balancer routes directly to the WebLogic Server, requests move from the load balancer to the application tier in one single HTTP jump, which can cause security concerns.
- The web tier allows the WebLogic Server cluster membership to be reconfigured (new servers added, others removed) without having to change the web server configuration (as long as at least some of the servers in the configured list remain alive).
- Oracle HTTP Server delivers static content more efficiently and faster than WebLogic Server; it also provides the ability to create virtual hosts and proxies via the Oracle HTTP Server configuration files.
- The web tier provides HTTP redirection over and above what the WebLogic Server provides. You can use Oracle HTTP Server as a front end against many different WebLogic Server clusters, and in some cases, control the routing by using content-based routing.
- Oracle HTTP Server provides the ability to integrate single sign-on capabilities into your enterprise deployment. For example, you can later implement single sign-on for the enterprise deployment by using Oracle Access Manager, which is part of the Oracle Identity and Access Management family of products.

- A web tier with Oracle HTTP Server provides support for WebSocket connections deployed within the WebLogic Server.

For more information about Oracle HTTP Server, see Introduction to Oracle HTTP Server in *Administering Oracle HTTP Server*.

Alternatives to Using a Web Tier

Although a Web tier provides a variety of benefits in an enterprise topology, Oracle also supports routing requests directly from the hardware load balancer to the Managed Servers in the middle tier.

This approach provide the following advantages:

- Lower configuration and processing overhead than using a front-end Oracle HTTP Server Web tier front-end.
- Monitoring at the application level since the LBR can be configured to monitor specific URLs for each Managed Server (something that is not possible with Oracle HTTP Server).

  You can potentially use this load balancer feature to monitor SOA composite application URLs. Note that this enables routing to the Managed Servers only when all composites are deployed, and you must use the appropriate monitoring software.

Configuration of Oracle HTTP Server in the Web Tier

Starting with Oracle Fusion Middleware 12*c*, the Oracle HTTP Server software can be configured in one of two ways: as part of an existing Oracle WebLogic Server domain or in its own standalone domain. Each configuration offers specific benefits.

When you configure Oracle HTTP Server instances as part of an existing WebLogic Server domain, you can manage the Oracle HTTP Server instances, including the wiring of communications between the web servers and the Oracle WebLogic Server Managed Servers by using Oracle Enterprise Manager Fusion Middleware Control. When you configure Oracle HTTP Server in a standalone configuration, you can configure and manage the Oracle HTTP Server instances independently of the application tier domains.

For this enterprise deployment guide, the Oracle HTTP Server instances are configured as separate standalone domains, one on each Web tier host. You can choose to configure the Oracle HTTP Server instances as part of the application tier domain, but this enterprise deployment guide does not provide specific steps to configure the Oracle HTTP Server instances in that manner.

See About Oracle HTTP Server in *Installing and Configuring Oracle HTTP Server*.

> **Note:**
>
> As of Fusion Middleware 12.2.1.4.0, Oracle Traffic Director has been deprecated. For an enterprise deployment, use Oracle HTTP Server.

About Mod_WL_OHS

As shown in the diagram, the Oracle HTTP Server instances use the WebLogic Proxy Plug-In (`mod_wl_ohs`) for proxying HTTP requests from Oracle HTTP Server to the Oracle WebLogic Server Managed Servers in the Application tier.

See What are Oracle WebLogic Server Proxy Plug-Ins? in *Using Oracle WebLogic Server Proxy Plug-Ins*.

About the Application Tier

The application tier consists of two physical host computers, where Oracle WebLogic Server and the Oracle Fusion Middleware products are installed and configured. The application tier computers reside in the secured zone between firewall 1 and firewall 2.

The following topics provide more information:

- Configuration of the Administration Server and Managed Servers Domain Directories
- Using Oracle Web Services Manager in the Application Tier
- Best Practices and Variations on the Configuration of the Clusters and Hosts on the Application Tier
- About the Node Manager Configuration in a Typical Enterprise Deployment
- About Using Unicast for Communications within the Application Tier
- About OPSS and Requests to the Authentication and Authorization Stores

Configuration of the Administration Server and Managed Servers Domain Directories

Unlike the Managed Servers in the domain, the Administration Server uses an active-passive high availability configuration. This is because only one Administration Server can be running within an Oracle WebLogic Server domain.

In the topology diagrams, the Administration Server on HOST1 is in the active state and the Administration Server on HOST2 is in the passive (inactive) state.

To support the manual fail over of the Administration Server in the event of a system failure, the typical enterprise deployment topology includes:

- A Virtual IP Address (VIP) for the routing of Administration Server requests.
- The configuration of the Administration Server domain directory on a shared storage device.

In the event of a system failure (for example a failure of HOST1), you can manually reassign the Administration Server VIP address to another host in the domain, mount the Administration Server domain directory on the new host, and then start the Administration Server on the new host.

However, unlike the Administration Server, there is no benefit to storing the Managed Servers on shared storage. In fact, there is a potential performance impact when Managed Server configuration data is not stored on the local disk of the host computer.

As a result, in the typical enterprise deployment, after you configure the Administration Server domain on shared storage, a copy of the domain configuration is placed on the local storage device of each host computer, and the Managed Servers are started from this copy of the domain configuration. You create this copy by using the Oracle WebLogic Server pack and unpack utilities.

The resulting configuration consists of separate domain directories on each host: one for the Administration Server (on shared storage) and one for the Managed Servers (on local storage). Depending upon the action required, you must perform configuration tasks from one domain directory or the other.

For more information about structure of the Administration Server domain directory and the Managed Server domain directory, as well as the variables used to reference these directories, see Understanding the Recommended Directory Structure for an Enterprise Deployment.

There is an additional benefit to the multiple domain directory model. It allows you to isolate the Administration Server from the Managed Servers. By default, the primary enterprise deployment topology assumes the Administration Server domain directory is on one of the application tier hosts, but if necessary, you could isolate the Administration Server further by running it from its own host, for example in cases where the Administration Server is consuming high CPU or RAM. Some administrators prefer to configure the Administration Server on a separate, dedicated host, and the multiple domain directory model makes that possible.

Using Oracle Web Services Manager in the Application Tier

Oracle Web Services Manager (Oracle WSM) provides a policy framework to manage and secure web services in the Enterprise Deployment topology.

In most enterprise deployment topologies, the Oracle Web Services Manager Policy Manager runs on Managed Servers in a separate cluster, where it can be deployed in an active-active highly available configuration.

You can choose to target Oracle Web Services Manager and Fusion Middleware products or applications to the same cluster, as long as you are aware of the implications.

The main reasons for deploying Oracle Web Services Manager on its own managed servers is to improve performance and availability isolation. Oracle Web Services Manager often provides policies to custom web services or to other products and components in the domain. In such a case, you do not want the additional Oracle Web Services Manager activity to affect the performance of any applications that are sharing the same managed server or cluster as Oracle Web Services Manager.

The eventual process of scaling out or scaling up is also better addressed when the components are isolated. You can scale out or scale up only the Fusion Middleware application Managed Servers where your products are deployed or only the Managed Servers where Oracle Web Services Manager is deployed, without affecting the other product.

Best Practices and Variations on the Configuration of the Clusters and Hosts on the Application Tier

In a typical enterprise deployment, you configure the Managed Servers in a cluster on two or more hosts in the application tier. For specific Oracle Fusion Middleware products, the enterprise deployment reference topologies demonstrate best practices for the number of Managed Servers, the number of clusters, and what services are targeted to each cluster.

These best practices take into account typical performance, maintenance, and scale-out requirements for each product. The result is the grouping of Managed Servers into an appropriate set of clusters within the domain.

Variations of the enterprise deployment topology allow the targeting of specific products or components to additional clusters or hosts for improved performance and isolation.

For example, you can consider hosting the Administration Server on a separate and smaller host computer, which allows the FMW components and products to be isolated from the Administration Server.

These variations in the topology are supported, but the enterprise deployment reference topology uses the minimum hardware resources while keeping high availability, scalability and security in mind. You should perform the appropriate resource planning and sizing, based on the system requirements for each type of server and the load that the system needs to sustain. Based on these decisions, you must adapt the steps to install and configure these variations accordingly from the instructions presented in this guide.

About the Node Manager Configuration in a Typical Enterprise Deployment

Starting with Oracle Fusion Middleware 12*c*, you can use either a per domain Node Manager or a per host Node Manager. The following sections of this topic provide more information on the impact of the Node Manager configuration on a typical enterprise deployment.

> **Note:**
>
> For general information about these two types of Node Managers, see Overview in *Administering Node Manager for Oracle WebLogic Server*.

About Using a Per Domain Node Manager Configuration

In a per domain Node Manager configuration—as opposed to a per host Node Manager configuration—you actually start two Node Manager instances on the Administration Server host: one from the Administration Server domain directory and one from the Managed Servers domain directory. In addition, a separate Node Manager instance runs on each of the other hosts in the topology.

The Node Manager that controls the Administration Server uses the listen address of the virtual host name created for the Administration Server. The Node Manager that controls the Managed Servers uses the listen address of the physical host. When the Administration Server fails over to another host, an additional instance of Node Manager is started to control the Administration Server on the failover host.

The key advantages of the per domain configuration are an easier and simpler initial setup of the Node Manager and the ability to set Node Manager properties that are unique to the Administration Server. This last feature was important in previous releases because some features, such as Crash Recovery, applied only to the Administration Server and not to the Managed servers. In the current release, the Oracle SOA Suite products can be configured for Automated Service Migration, rather than Whole Server Migration. This means the Managed Servers, as well as the Administration Server, can take advantage of Crash Recovery, so there is no need to apply different properties to the Administration Server and Managed Server domain directories.

Another advantage is that the per domain Node Manager provides a default SSL configuration for Node Manager-to-Server communication, based on the Demo Identity store created for each domain.

About Using a Per Host Node Manager Configuration

In a per host Node Manager configuration, you start a single Node Manager instance to control the Administration Server and all Managed Servers on a host, even those that reside in different domains. This reduces the footprint and resource utilization on the Administration Server host, especially in those cases where multiple domains coexist on the same computer.

A per host Node Manager configuration allows all Node Managers to use a listen address of ANY, so they listen on all addresses available on the host. This means that when the Administration Server fails over to a new host, no additional configuration is necessary. The per host configuration allows for simpler maintenance, because you can update and maintain a single Node Manager properties file on each host, rather than multiple node manager property files.

The per host Node Manager configuration requires additional configuration steps. If you want SSL for Node Manager-to-Server communication, then you must configure an additional Identity and Trust store, and it also requires using Subject Alternate Names (SAN), because the Node Manager listens on multiple addresses. Note that SSL communications are typically not required for the application tier, because it is protected by two firewalls.

About Using Unicast for Communications within the Application Tier

Oracle recommends the unicast communication protocol for communication between the Managed Servers and hosts within the Oracle WebLogic Server clusters in an enterprise deployment. Unlike multicast communication, unicast does not require cross-network configuration and it reduces potential network errors that can occur from multicast address conflicts as well.

When you consider using the multicast or unicast protocol for your own deployment, consider the type of network, the number of members in the cluster, and the reliability requirements for cluster membership. Also consider the following features of each protocol.

Features of unicast in an enterprise deployment:

- Uses a group leader that every server sends messages directly to. This leader is responsible for retransmitting the message to every other group member and other group leaders, if applicable.
- Works out of the box in most network topologies
- Requires no additional configuration, regardless of the network topology.
- Uses a single missed heartbeat to remove a server from the cluster membership list.

Features of multicast in an enterprise deployment:

- Multicast uses a more scalable peer-to-peer model, where a server sends each message directly to the network once and the network makes sure that each cluster member receives the message directly from the network.
- Works out of the box in most modern environments, where the cluster members are in a single subnet.
- Requires additional configuration in the routers and WebLogic Server (that is, Multicast TTL) if the cluster members span more than one subnet.
- Uses three consecutive missed heartbeats to remove a server from the cluster membership list.

Depending on the number of servers in your cluster and on whether the cluster membership is critical for the underlying application (for example, in session-replication intensive applications or clusters with intensive RMI invocations across the cluster), each model may act better.

Consider whether your topology is going to be part of an active-active disaster recovery system or if the cluster is going to traverse multiple subnets. In general, unicast acts better in those cases.

For more information about multicast and unicast communication types, see the following resources:

- Configuring Multicast Messaging for WebLogic Server Clusters in *High Availability Guide*

- One-to-Many Communication Using Unicast in *Administering Clusters for Oracle WebLogic Server*

About OPSS and Requests to the Authentication and Authorization Stores

Many of the Oracle Fusion Middleware products and components require an Oracle Platform Security Services (OPSS) security store for authentication providers (an identity store), policies, credentials, keystores, and for audit data. As a result, communications must be enabled so the application tier can send requests to and from the security providers.

For authentication, this communication is to an LDAP directory, such as Oracle Unified Directory (OUD), which typically communicates over port 389 or 636. When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server Authentication provider. However, for an enterprise deployment, you must use a dedicated, centralized LDAP-compliant authentication provider.

For authorization (and the policy store), the location of the security store varies, depending upon the tier:

- For the application tier, the authorization store is database-based, so frequent connections from the Oracle WebLogic Server Managed Servers to the database are required for the purpose of retrieving the required OPSS data.
- For the web tier, the authorization store is file-based, so connections to the database are not required.

For more information about OPSS security stores, see the following sections of *Securing Applications with Oracle Platform Security Services*:

- Authentication Basics
- The Security Model

About the Data Tier

In the data tier, an Oracle RAC database runs on the two hosts (DBHOST1 and DBHOST2). The database contains the schemas required by the Oracle Identity and Access Management components and the Oracle Platform Security Services (OPSS) policy store.

You can define multiple services for the different products and components in an enterprise deployment to isolate and prioritize throughput and performance accordingly. In this guide, one database service is used as an example. Furthermore, you can use other high availability database solutions to protect the database:

- Oracle Data Guard: See Introduction to Oracle Data Guard in *Oracle Data Guard Concepts and Administration*.
- Oracle RAC One Node: See Overview of Oracle RAC One Node in *Oracle Real Application Clusters Administration and Deployment Guide*.

These solutions above provide protection for the database beyond the information provided in this guide, which focuses on using an Oracle RAC Database, given the scalability and availability requirements that typically apply to an enterprise deployment.

For more information about using Oracle Databases in a high availability environment, see Database Considerations in *High Availability Guide*.

3 About the IAM Enterprise Deployment

Learn about deploying Oracle Identity and Access Management topologies on commodity hardware. These topologies represent specific reference implementations of the concepts described in About a Typical Enterprise Deployment.
This chapter includes the following topics:

- About the Primary and Build-Your-Own Enterprise Deployment Topologies
  There are two primary reference topologies for Oracle Identity and Access Management. While the components installed into each topology are the same, the exact Oracle Identity and Access Management topology you install and configure for your organization may vary.
- Diagram of Oracle Identity and Access Management on Distributed Hardware
- About the Primary Oracle Identity and Access Management Topology Diagrams
  The primary enterprise deployment topology is the main Oracle reference topology for Oracle Identity and Access Management. It provides a solution which is both highly available and scalable.
- About the Forgotten Password Functionality
  In Oracle 11*g*, the mechanism to reset passwords was provided by Oracle Identity Governance. In Oracle 12*c*, you have two possibilities - by integrating Oracle Access Management (OAM) and Oracle Identity Governance (OIG) or by using Oracle Access Management.
- Integrating Oracle LDAP, Oracle Access Manager, and Oracle Identity Governance
  Integration of Oracle Identity Manager and Oracle Access Manager with LDAP directories is done by using LDAP Connector.
- Roadmap for Implementing the Primary IAM Suite Topologies
- Building Your Own Oracle Identity and Access Management Topology
  These step-by-step instructions help you configure the two primary enterprise topologies for Oracle Identity and Access Management. However, Oracle recognizes that the requirements of your organization may vary, depending on the specific set of Oracle Fusion Middleware products you purchase and the specific types of applications you deploy.
- About Using Service or Server Migration to Enable High Availability of the Enterprise Topology

About the Primary and Build-Your-Own Enterprise Deployment Topologies

There are two primary reference topologies for Oracle Identity and Access Management. While the components installed into each topology are the same, the exact Oracle Identity and Access Management topology you install and configure for your organization may vary.

This guide provides step-by-step instructions for installing and configuring these topologies.

The components installed into each topology are the same. The difference being that one deployment is concentrated onto a small number of highly specified servers and the second is distributed amongst a larger number of smaller machines.

To simplify the installation and configuration process, this guide utilizes the IAM deployment wizard, which once you tell it how to layout your topology, will automatically configure it for you.

After you have created your deployment, this guide will show you how to extend it to include additional IAM products, which you may want to use. The procedures in this book do not cover every IAM product. The steps in this guide can easily be adapted to any other IAM product you may want to include.

Diagram of Oracle Identity and Access Management on Distributed Hardware

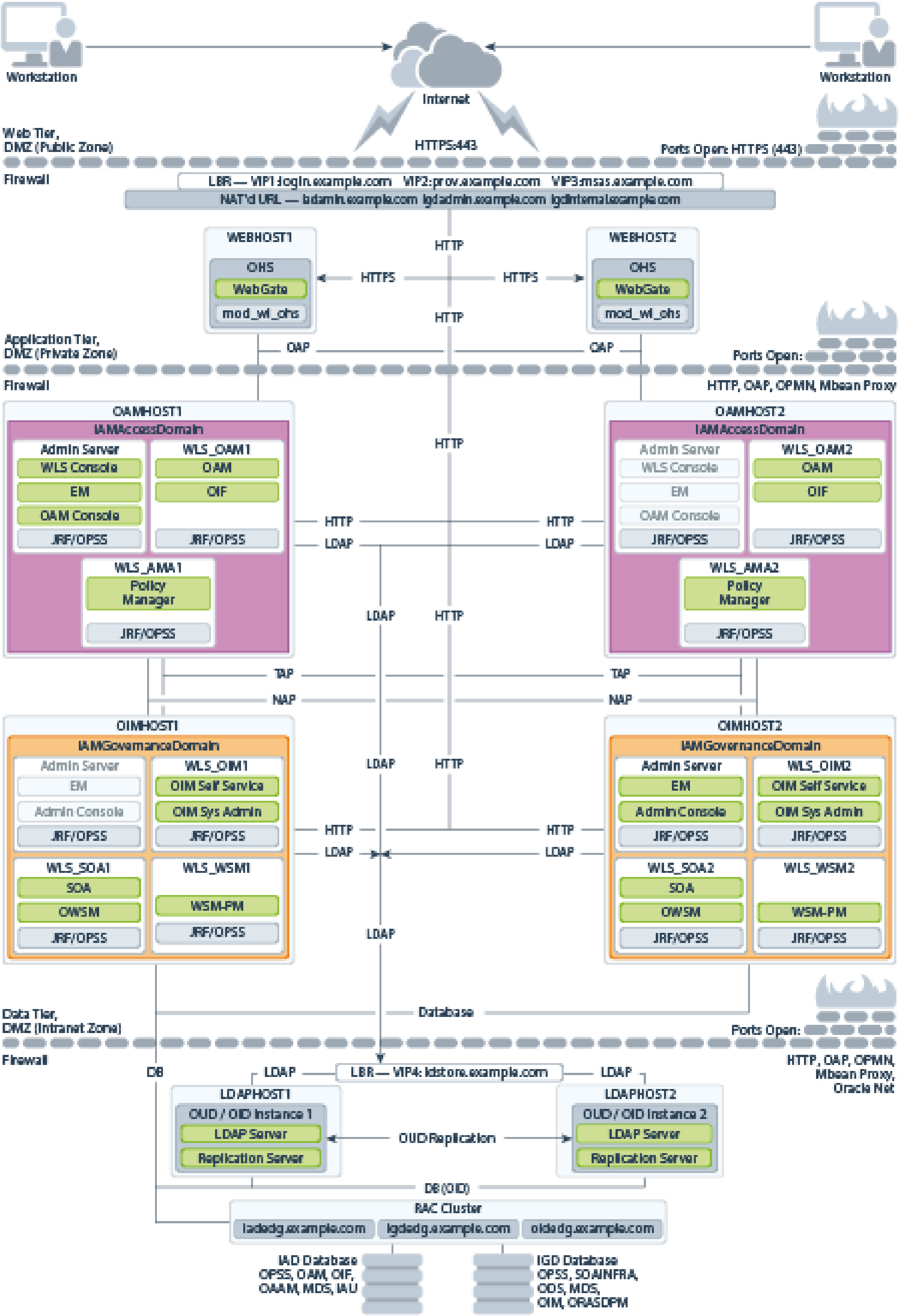
The sample topology in this section shows the eight-node distributed topology.

In this topology, the software is distributed across eight hosts: two hosts for the web tier and four hosts for the application tier, and two hosts for the directory services.

This topology is a typical deployment if you are using virtual machines (VMs) that are less powerful, but easy to create and manage. You deploy key components, such as Oracle Access Management, Oracle Identity Governance, and Directory Services on their own dedicated hosts.

The following illustration shows the Oracle Identity and Access Management topology. The diagram is shown with complete separation of components. If you wish to use less hardware, then you can collocate the components as required. For information about the system requirements for each host, see Host Computer Hardware Requirements.

Figure 3-1 Oracle Identity and Access Management Topology



About the Primary Oracle Identity and Access Management Topology Diagrams

The primary enterprise deployment topology is the main Oracle reference topology for Oracle Identity and Access Management. It provides a solution which is both highly available and scalable.

- Product Separation
- Understanding the Directory Tier
- About Oracle Unified Directory Assured Replication
- Summary of Oracle Identity and Access Management Load Balancer Virtual Server Names
- Summary of the Managed Servers and Clusters on the Application Tier Hosts

Product Separation

An IAM deployment is made up of a number of components. These components include:

- Web Servers
- WebLogic Application Servers
- LDAP
- Database

The Oracle Identity and Access Management components are split into two different domains: IAMAccessDomain and IAMGovernanceDomain. Products are distributed as follows:

- IAMAccessDomain contains Oracle Access Management (OAM).
- IAMGovernanceDomain contains Oracle Identity Governance.
- OHSDomain used to host the Oracle HTTP Servers.
- OUDSMDomain used when OUDSM is deployed.

This split is due to the different operational and availability requirements demanded of the individual components. Typically components in the IAMAccessDomain have a higher availability requirement than those in the IAMGovernanceDomain. By separating these components out, you can manage the availability requirements differently. You can patch governance components independently of access components, and you can shut down the Governance instance without impacting the access components. From Oracle Identity and Access Management 12c, it is not supported to have Oracle Access Manager and Oracle Identity Governance in the same domain.

In addtion to the domains above, note that Oracle Unified Directory is deployed without a domain.

A further benefit of this separation is that you can build a topology in a modular fashion. You can start with a directory and extend it to Access components, then later extend it to Governance components, without needing to affect the deployed software or configuration of existing components, unless you are wiring them together.

Understanding the Directory Tier

The Directory tier consists of two or more physical host computers, where an LDAP compliant directory is installed. Typically, this is Oracle Unified Directory (OUD).

The Directory tier is often combined with the Data tier.

This release of the Enterprise Deployment Guide supports three different LDAP directories. You may be creating this directory for the first time, or you may be using existing directory from within the organization. The Oracle Unified Directory (OUD) directory is supported.

The directory you choose will be organization dependent.

About Oracle Unified Directory Assured Replication

Oracle Unified Directory server instances natively use replication to keep their embedded databases in sync. By default, replication employs a loose consistency model in which the updates are replicated to replicas AFTER returning the operation result to the application. In this model it is therefore possible to write some data to a replica, and read outdated information from another replica for a short time after the write. Great efforts have been made in Oracle Unified Directory replication to ensure that the replication process is fast and can achieve replication in the order of one millisecond.

Oracle Unified Directory can be configured to use the Assured Replication model, which has been developed to guarantee that the data in the replicas is consistent. When using the Safe Read mode of Assured Replication, applications have the guarantee that the replication process is completed before returning the result of a write operation.

Using Assured Replication has a negative impact on the response time of write operations because it requires some communications with remote replicas before returning the operation result. The amount of the delay varies, depending on the network being used and the capacity of the servers hosting Oracle Unified Directory. Using Assured replication has little if any impact on read operations.

If you expect to regularly perform large writes to your directory, consider configuring your load balancer to distribute requests to your Oracle Unified Directory instances in an active/passive mode. This will remove the chance of you reading out of date data from a replica, but could result in overall performance degradation if your Oracle Unified Directory host is not capable of processing all of the requests.

For the purposes of this Guide, it is assumed that the ability to have multiple servers processing requests is more important than the extra overhead incurred with writing requests in Assured mode. To that end, this Guide shows the configuration of Oracle Unified Directory using Assured Replication. Both of the following Oracle Unified Directory configurations, however, are supported:

- Active/Active in an assured configuration
- Active/Passive in a non assured configuration

For more information, see the Assured Replication section of *Oracle Fusion Middleware Administrator's Guide for Oracle Unified Directory*.

Summary of Oracle Identity and Access Management Load Balancer Virtual Server Names

In order to balance the load on servers and to provide high availability, a hardware load balancer is required. This hardware load balancer should exist on redundant hardware to ensure maximum availability. The hardware load balancer must be configured to recognize a set of virtual server names.

The hardware load balancer in Oracle Identity and Access Management deployments must recognize the following virtual server names.

- `login.example.com` - This virtual server name is used for all incoming Access traffic. It acts as the access point for all HTTP traffic to the runtime Access Management components. The load balancer routes all requests to this virtual server name over SSL. As a result, clients access this service using the following secure address:

  `login.example.com:443`

- `prov.example.com` - This virtual server name is used for all incoming Governance traffic. It acts as the access point for all HTTP traffic to the runtime Governance components. The load balancer routes all requests to this virtual server name over SSL. As a result, clients access this service using the following secure address:

  `prov.example.com:443`

  Note that, in previous releases of the Enterprise Deployment Guide, `login.example.com` and `prov.example.com` were the same entry point. This release allows for them to be separated out. This will enable smarter routing from the load balancer, allow a more modular deployment and will facilitate future Multi-datacenter deployments. If desired these two entry points can still be combined to provide a single point of entry into the IAM deployment.

- `iadadmin.example.com` - This virtual server name is enabled on the load balancer. It acts as the access point for all internal HTTP traffic that gets directed to the administration services in the IAMAccessDomain. The incoming traffic from clients is non-SSL enabled. Therefore, the clients access this service using the following address:

  `iadadmin.example.com:80`

  This in turn is forwarded to port `7777` on WEBHOST1 and WEBHOST2.

  The services accessed on this virtual host include the WebLogic Administration Server Console and Oracle Enterprise Manager Fusion Middleware Control.

  Create rules in the firewall to block outside traffic from accessing the /console and /em URLs using this virtual host. Only traffic inside the DMZ should be able to access these URLs on the iadadmin.example.com virtual host.

- `igdadmin.example.com` - This virtual server name is enabled on the load balancer. It acts as the access point for all internal HTTP traffic that gets directed to the administration services in the IAMGovernanceDomain. The incoming traffic from clients is non-SSL enabled. Therefore, the clients access this service using the following address:

  igdadmin.example.com:80

  This in turn is forwarded to port `7777` on WEBHOST1 and WEBHOST2.

  The services accessed on this virtual host include the WebLogic Administration Server Console and Oracle Enterprise Manager Fusion Middleware Control.

Create rules in the firewall to block outside traffic from accessing the /console and /em URLs using this virtual host. Only traffic inside the DMZ should be able to access these URLs on the igdadmin.example.com virtual host.

- `igdinternal.example.com` - This virtual server name is for internal communications between the application tier components in the Governance Domain only and is not exposed to the Internet. This virtual server is used for both Oracle OIM Suite and Oracle SOA Suite internal communications.

  The traffic from clients to this virtual server is not SSL-enabled. Clients access this service using the following address and the requests are forwarded to port 7777 on WEBHOST1 and WEBHOST2:

  ```
  igdinternal.example.com:7777
  ```

- `idstore.example.com` - This virtual server name is used for all incoming identity store traffic. It acts as the access point to the LDAP directory instances. This virtual server is not exposed to the internet.

  The traffic from clients to this virtual server may or may not be SSL-enabled, depending on the type of LDAP directory in use. Typically, this will be non-SSL enabled for Oracle Unified Directory. Clients access this service using this virtual server name and the requests are forwarded to the LDAP instances.

- `oam.example.com:5575` — This is an additional load balancer virtual host used in multi datacenter deployments only. This virtual host routes requests to the Oracle Access Management (OAM) proxy port on the OAM Managed servers. For example, `5575`.

Summary of the Managed Servers and Clusters on the Application Tier Hosts

The Application tier hosts the Administration Server and Managed Servers in the Oracle WebLogic Server domains.

Depending upon the components you select, the Oracle WebLogic Server domain for the Oracle Identity and Access Management consists of the clusters shown in Table 3-1. These clusters function as active-active high availability configurations.

Table 3-1 Domain Clusters and Managed Servers

Domain	Cluster	Managed Servers
IAMAccessDomain	Oracle Access Manager	oam_server1, oam_server2
	Oracle Policy Manager	oam_policy_mgr1, oam_policy_mgr2
IAMGovernanceDomain	Oracle Identity Governance	oim_server1, oim_server2
	Oracle SOA Suite	soa_server1, soa_server2
	Oracle Web Services Manager	WLS_WSM1, WLS_WSM2

About the Forgotten Password Functionality

In Oracle 11*g*, the mechanism to reset passwords was provided by Oracle Identity Governance. In Oracle 12*c*, you have two possibilities - by integrating Oracle Access Management (OAM) and Oracle Identity Governance (OIG) or by using Oracle Access Management.

- Oracle Access Management (OAM) and Oracle Identity Governance (OIG) integration:

  In this scenario, when the users first sign in, they can set a number of challenge questions stored in OIG. If users forget their password, they can answer the challenge questions. If they answer their challenge questions successfully, they will be given the option to reset their password by using OIG.

- Oracle Access Management:

  In this scenario, OAM is wired to the Oracle User Messaging Service. Each user is associated with either a telephone number and or an email address. When users request for a forgotten password link, they are sent a one time PIN which they can enter into the application to reset the password.

This guide describes how to set up both the scenarios.

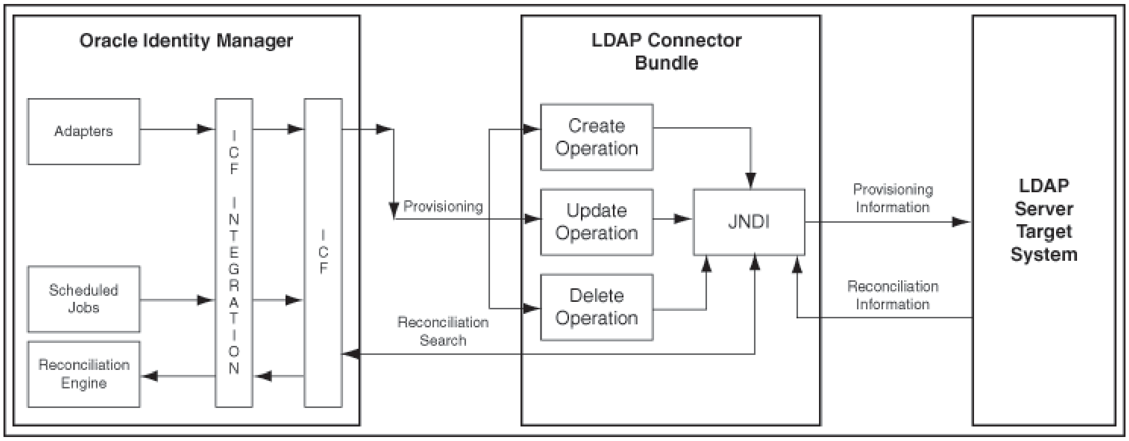
Integrating Oracle LDAP, Oracle Access Manager, and Oracle Identity Governance

Integration of Oracle Identity Manager and Oracle Access Manager with LDAP directories is done by using LDAP Connector.

To enable termination of user sessions upon disablement or termination of a user, download the 12.2.1.3 version of the LDAP Connector.

This section describes how to obtain, install, and configure the Oracle Connector for LDAP.

About the Oracle Connector



The LDAP Connector is implemented by using the Identity Connector Framework (ICF). The ICF is a component that provides basic reconciliation and provisioning operations that are common to all Oracle Identity Manager connectors. The ICF is shipped along with Oracle Identity Manager. Therefore, you need not configure or modify the ICF.

The LDAP Connector uses JNDI to access the target system. This connector can be configured to run in one of the following modes:

- **Identity Reconciliation**: Identity reconciliation is also known as authoritative or trusted source reconciliation. In this form of reconciliation, OIM Users are created or updated corresponding to the creation of and updates to users on the target system.

> **Note:**
>
> The identity reconciliation mode supports the reconciliation of user objects only.

- **Account Management**: Account management is also known as target resource management. This mode of the connector enables provisioning and target resource reconciliation.

Provisioning

Provisioning involves creating, updating, or deleting users, groups, roles, and organizational units (OUs) on the target system through Oracle Identity Manager.

When you allocate (or provision) a target system resource to an OIM user, the operation results in the creation of an account on the target system for that user. In the Oracle Identity Manager context, the term **provisioning** is also used to mean updates (for example enabling or disabling) made to the target system account through Oracle Identity Manager.

Users and organizations are organized in a hierarchical format on the target system. Before you can provision users to (that is, create users in) the required organizational units (OUs) on the target system, you must fetch into Oracle Identity Manager the list of OUs used on the target system. This is achieved by using the LDAP Connector OU lookup Reconciliation scheduled job for lookup synchronization.

Similarly, before you can provision users to the required groups or roles on the target system, you must fetch into Oracle Identity Manager the list of all groups and roles used on the target system. This is achieved by using the LDAP Connector Group Lookup Reconciliation and LDAP Connector Role Lookup Recon scheduled jobs for lookup synchronization.

Target resource reconciliation

To perform target resource reconciliation, the LDAP Connector User Search Reconciliation or LDAP Connector User Sync Reconciliation scheduled jobs is used. The connector applies filters to locate users to be reconciled from the target system and then fetches the attribute values of these users.

Depending on the data that you want to reconcile, you use different scheduled jobs. For example, you use the LDAP Connector User Search Reconciliation scheduled job to reconcile user data in the target resource mode.

You can deploy the Oracle LDAP Connector either locally in Oracle Identity Manager or remotely in the connector server. A **connector server** enables remote execution of an Identity Connector. This guide explains the steps to install and configure the connector locally in Oracle Identity Manager.

Roadmap for Implementing the Primary IAM Suite Topologies

Table 3-2 provides a roadmap for implementing the primary IAM suite topologies on commodity hardware.

Table 3-2 Roadmap for Implementing Primary IAM Suite Topologies on Commodity Hardware

Scenario	Tasks	For More Information, See
Creating an IAM Enterprise Deployment manually on commodity hardware	Understand a typical enterprise deployment and review the primary deployment topologies.	Enterprise Deployment Overview About a Typical Enterprise Deployment About the IAM Enterprise Deployment About a Multi-Data Center Deployment
	Review the hardware and software requirements and procure the resources for the enterprise deployment.	Procuring Resources for an Enterprise Deployment
	Prepare the load balancers and firewalls.	Preparing the Load Balancer and Firewalls for an Enterprise Deployment
	Prepare the storage and understand the directory structure.	Preparing the File System for an Enterprise Deployment
	Configure the host computers.	Preparing the Host Computers for an Enterprise Deployment
	Prepare the database.	Preparing the Database for an Enterprise Deployment
	Configure Oracle LDAP.	Configuring Oracle LDAP for an Enterprise Deployment
	Create the Oracle Fusion Middleware Infrastructure for Oracle Access Management.	Creating Infrastructure for Oracle Access Management
	Create the Oracle Fusion Middleware Infrastructure for Oracle Identity Governance.	Creating Infrastructure for Oracle Identity Governance
	Configure Oracle HTTP Server	Configuring Oracle HTTP Server for an Enterprise Deployment
	Configure Oracle Access Management (OAM).	Configuring Oracle Access Management
	Configure Oracle Identity Governance (OIG) or (OIM).	Configuring Oracle Identity Governance
	Configure server migration settings.	Using Whole Server Migration and Service Migration in an Enterprise Deployment
	Configure Single Sign-On (SSO).	Configuring Single Sign-On for an Enterprise Deployment

Building Your Own Oracle Identity and Access Management Topology

These step-by-step instructions help you configure the two primary enterprise topologies for Oracle Identity and Access Management. However, Oracle recognizes that the requirements of your organization may vary, depending on the specific set of Oracle Fusion Middleware products you purchase and the specific types of applications you deploy.

For information on the primary enterprise topologies for Oracle Identity and Access Management, see Figure 3-1.

In many cases, you can install and configure an alternative topology, one that includes additional components, or one that does not include all the Oracle Identity and Access Management products shown in the primary topology diagrams.

A few alternatives to the reference Oracle Identity and Access Management topologies you can implement by using the steps provided in this guide are:

- OAM Only with an Existing Directory
- OIM Only with an Existing Directory
- OAM/OIM Integrated in a Modular Deployment
- OAM/OIM Integrated with an Existing Directory

About Using Service or Server Migration to Enable High Availability of the Enterprise Topology

To ensure high availability of the Oracle Identity Governance products and components, this guide recommends that you enable Oracle WebLogic Server Whole Server Migration for the Oracle Identity Manager and Oracle SOA Suite clusters that you create as part of the reference topology.

For **static clusters**, you can configure Automatic Service Migration by using the Configuration Wizard HA Options screen. If you select the **Enable Automatic Service Migration** option, it configures migratable target definitions that are required for automatic service migration. In the same screen, you can use **JTA Transaction Log Persistence** and **JMS Server Persistence** options to configure them with JDBC stores automatically. Oracle recommends that you enable these options when you configure static clusters in the OIM enterprise deployment.

For **dynamic clusters**, migratable targets are not required because some functionalities of the automatic service migration are provided inherently by the dynamic cluster. The Configuration Wizard High Availability Options screen is not used for dynamic clusters but some additional steps are required to configure the leasing and the persistent stores migration policies. Oracle recommends that you perform these post-steps when you configure dynamic clusters in the OIM enterprise deployment.

For more information, see Using Whole Server Migration and Service Migration in an Enterprise Deployment.

4

About a Multi-Data Center Deployment

This chapter describes the multi-data center active-passive or active-passive disaster protection and the multi-data center active-active or active-active disaster protection.

Traditional disaster protection systems, also called multi-data center active-passive or active-passive disaster protection, use a model where one site is running while another site is on standby, to prevent possible failover scenarios. These disaster protection systems usually incur increased operational and administration costs, while the need for continuous use of resources and increased throughput (that is, avoiding situations where the standby machines are idle) have increased through the years.

IT systems’ design is increasingly driven by capacity utilization and even distribution of load, which leads to the adoption of disaster protection solutions that use, all the resources available, as much as possible. These disaster protection systems are called multi-data center active-active or active-active disaster protection.

A Multi-Data Center deployment for Oracle Identity and Access Management is explained in the following topics:

- About the Oracle Identity and Access Management Multi-Data Center Deployment
- Administering Oracle Identity and Access Management Multi-Data Center Deployment
- About the Requirements for Multi-Data Center Deployment
- About the Characteristics of a Multi-Data Center Deployment

About the Oracle Identity and Access Management Multi-Data Center Deployment

Oracle Identity and Access Management consists of two products, each of which has a different availability requirements:

- Oracle Access Management (OAM), which is used to collect credentials and grant access to other system resources.
- Oracle Identity Governance (OIG), which has the ability to create new accounts and grant access rights to those accounts.

Many organizations use both products, which they integrate together to provide a complete solution. However, the approach to achieve an Active/Active deployment in each case is different. OAM has had a multi-data center solution since 11*g* Release 2 (11.1.2.2.0), but this has been a standalone solution, which does have the ability to integrate with OIG. Integration between the products has to be handled carefully to maximize the benefits of the two different solutions.

- About Multi-Data Center Deployment Approach in OAM and OIG
- About the Multi-Data Center Deployment in OAM and OIG

About Multi-Data Center Deployment Approach in OAM and OIG

The two product groups, Oracle Access Management (OAM) and Oracle Identity Governance (OIG) need to be treated differently:

- OAM uses two independent databases and proprietary OAM replication technologies to keep those databases in sync.
- OIG has no dedicated multi-data center (MDC) solution and therefore must use a solution similar to that of the traditional disaster recovery solution. That is, this solution uses a single database, which is replicated, to the disaster recovery site. All writes to the database goes to the site, whichever is active.

When a failure occurs in the OIG database tier, both multi-data center deployment active-active and multi-data center active-passive present similar Recovery Time Objective (RTO) and Recovery Point Objective (RPO), since the database is the driver for recovery. In both the cases, the database is active only in one Site and passive in the other site.

The only advantage of multi-data center active-active deployment systems is that an appropriate data source configuration can automate the failover of database connections from the middle tiers, reducing RTO (the recovery time is decreased because restart of the middle tiers is not required).

When a failure occurs in the OAM database tier, there can be no discernible system impact, as the surviving database continues to process requests. When configuring an OAM multi-data center deployment, you tell the system how you wish failover to be handled. The options are:

- Continue as if nothing has happened; Site–2 will accept authentications from Site–1.
- Force sessions to be re-authenticated on Site–2.

When the OAM solution is active-active, the OIG solution is usually active-passive.

Whilst OAM and OIG can be treated differently using the OAM MDC solution and OIG using the traditional stretched cluster active-passive solution. They can be treated the same, that is to say using a single active database replicated to a second site using active dataguard, and a Stretched Weblogic Cluster spanning both sites. A single solution is easier to maintain and manage, however if active-active is required, the two sites must be close together. Using the OAM MDC approach combined with OIG running active-passive allows the two sites to be much further apart.

However, note that, even in a distributed OAM active-active solution, OAM policy changes can only be made on one instance, which is designated the primary instance.

If you are using a stretched cluster deployment for OIG or OAM, then, whether you can use those sites simultaneously or in an active-passive manner or not, is determined by the speed of the network. The higher the network latency between the two sites, the performance degrades. In a stretched cluster deployment, a network latency of greater than 5ms is normally considered to be unacceptable. This figure is a guide to only many things, including transaction volumes, which affects the minimum value. If you are using a stretched cluster in an active-active deployment, suitable load testing should be conducted to ensure that the performance is acceptable for your given topology.

About the Multi-Data Center Deployment in OAM and OIG

Besides the common performance paradigms that apply to single-datacenter designs, Oracle Identity and Access Management multi-data center active-active systems need to minimize the traffic across sites to reduce the effect of latency on the system's throughput. In a typical Oracle Identity and Access Management system, besides database access (for dehydration, metadata access, and other database read/write operations that custom services that participate in the system may perform), communication between the different tiers can occur mainly over the following protocols:

- Incoming HTTP invocations from load balancers (LBR) or Oracle Web Servers (Oracle HTTP Server) and HTTP callbacks
- Incoming HTTP invocations between OAM and OIM
- Java Naming and Directory Interface (JNDI)/Remote Method Invocation (RMI) and Java Message Service (JMS) invocations between Oracle WebLogic Servers
- Oracle Access Protocol (OAP) requests between Web Servers and OAM

For improved performance, all of the above protocols should be restrained, as much as possible, to one single site. That is, servers in SiteN ideally should just receive invocations from Oracle Web Servers in SiteN. These servers should make JMS, RMI and JNDI invocations only to servers in SiteN and should get callbacks generated by servers only in SiteX. Additionally, servers should use storage devices that are local to their site to eliminate contention (latency for NFS writes across sites may cause severe performance degradation). Only if a component is available in SiteN, a request be sent to an alternate site.

There are additional types of invocations that may take place between the different Oracle Identity and Access Management servers that participate in the topology. These invocations are:

- Oracle Coherence notifications: Oracle Coherence notifications need to reach all servers in the system to provide a consistent compoSite and metadata image to all SOA requests, whether served by one site or the other.
- HTTP session replications: Some Oracle Identity and Access Management components use stateful web applications that may rely on session replication to enable transparent failover of sessions across servers. Depending on the usage patterns and number of users, considerable amount of replication data may be generated. Replication and failover requirements have to be analyzed for each business case, but ideally, session replication traffic should be reduced across sites as much as possible
- LDAP or policy or identity store access: Access to policy and identity stores is performed by Oracle WebLogic Server infrastructure and Oracle Identity and Access Managementcomponents, for authorization and authentication purposes. In order to enable seamless access to users from either site, a common policy or identity store view needs to be used. Ideally, each site should have an independent identity and policy store that is synchronized regularly to minimize invocations from one site to the other.

Administering Oracle Identity and Access Management Multi-Data Center Deployment

A key aspect of the design and deployment of an Oracle Identity and Access Management multi-data center deployment is the administration overhead introduced by the solution.

In order to keep a consistent reply to requests, the sites involved should use a configuration such that the functional behavior of the system is the same irrespective of which site is processing those requests. Oracle Identity and Access Management keeps its configuration and metadata in the Oracle pdatabase. Hence, multi-data center active-active deployments with a unique active database guarantee consistent behavior at the composite and metadata level (there is a single source of truth for the involved artifacts).

The Oracle WebLogic Server configuration, however, is kept synchronized across multiple nodes in the same domain by the Oracle WebLogic Server Infrastructure. Most of this configuration usually resides under the Administration Server’s domain directory. This configuration is propagated automatically to the other nodes in the same domain that contain Oracle WebLogic Servers. Based on this, the administration overhead of a multi-data center active-active deployment system is very small as compared to any active-passive approach, where constant replication of configuration changes is required.

Oracle Fusion Middleware binaries across all sites must be the same, should be at the same location, and the same patches should be applied. This can be achieved by independent installation or by disk mirroring. If you are using disk mirroring, ensure that at least two different versions are available so that a corrupt patch only impacts half of the deployment (in a site) and the binary corruption is not replicated to the disaster recovery site. For example:

- FMW binary set 1 – SiteAHost1, SiteAHost3, SiteAHost5
- FMW binary set 2 – SiteAHost2, SiteAHost4, SiteAHost6

FMW binary set 1 and 2 replicated to Site–2 and mounted to:

- FMW binary set 1 (copy) – SiteBHost1, SiteBHost3, SiteBHost5
- FMW binary set 2 (copy) – SiteBHost2, SiteBHost4, SiteBHost6

About the Requirements for Multi-Data Center Deployment

The requirements to set up an Oracle Identity and Access Management multi-data center deployment are:

- Toplogy
- Entry points
- Database
- Directory Tier

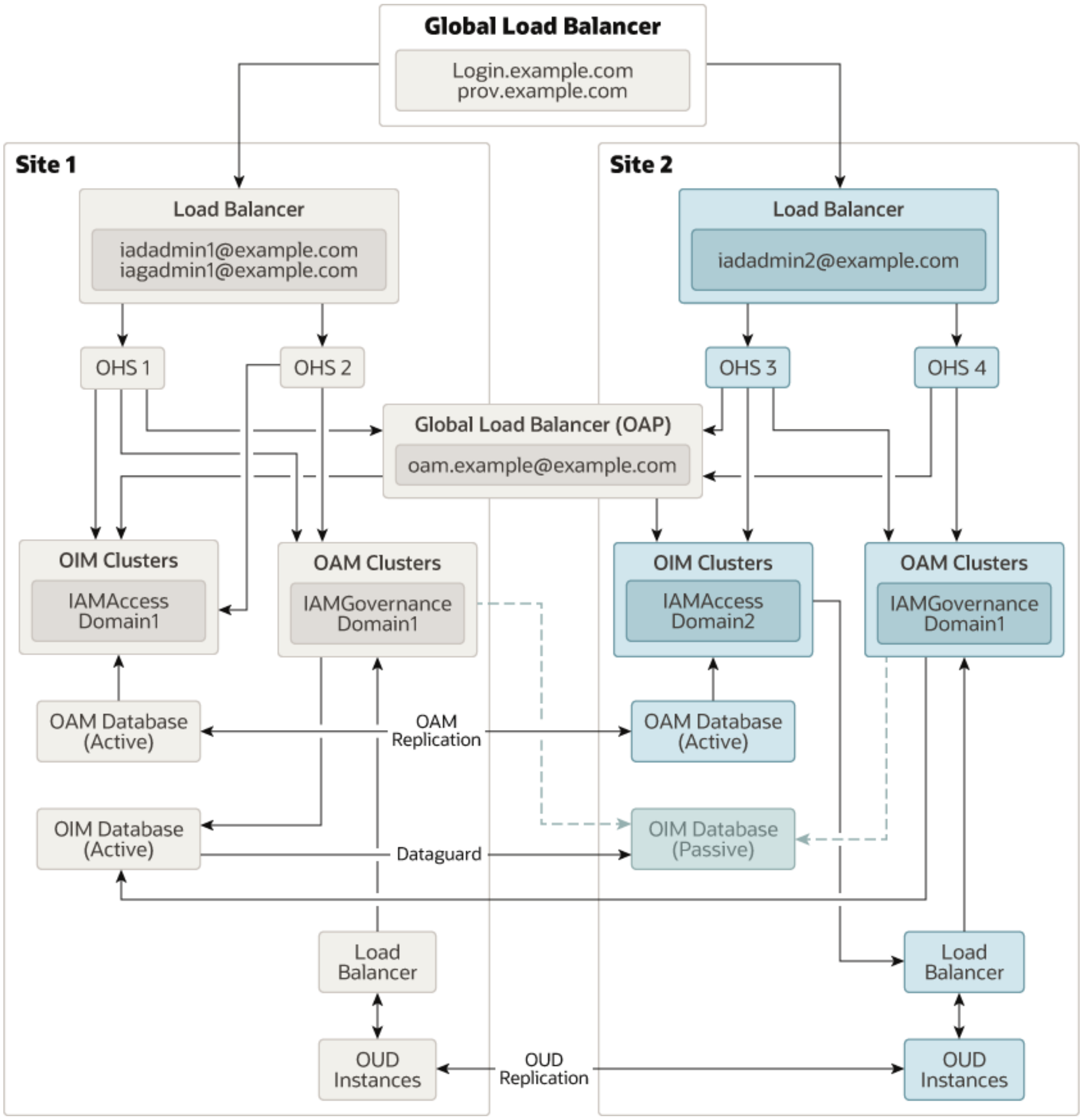
These requirements are explained in the following sections.

- About the Multi-Data Center Deployment Topology
- About the Entry Points in Multi-Data Center Deployment
- About the Databases in Multi-Data Center Deployment
- About the Directory Tier in Multi-Data Center Deployment
- About the Load Balancers in Multi-Data Center Deployment
- Shared Storage Versus Database for Transaction Logs and Persistent stores

About the Multi-Data Center Deployment Topology

Review the topology model for an Oracle Identity and Access Management active-active multi-data center deployment.

The analysis and recommendations included are based on the topology described in this section. Each site locally uses a slightly modified version of the Oracle Identity and Access Management enterprise deployment Topology. For more information, refer the following sections.



In the Oracle Identity and Access Management active-active multi-data center deployment topology:

- There are two separate sites (Site–1 and Site–2 for future reference in this document) that are accessed by one unique access point:
  - A global load balancer, which directs traffic to either site (each vendor provides different routing algorithms).
  - A local load balancer, the local access point of each site, which distributes requests to multiple Oracle HTTP Server (OHS) that, in turn, allocates requests to specific Oracle WebLogic Servers hosting Identity and Access Management components.
- Each Oracle Access Management (OAM) implementation accesses a local database which is read/write, the databases are kept in sync using OAM replication.
- The Oracle Identity Governance (OIG) implementation shares one unique database that is accessed concurrently by servers in both sites (if they are both active).
- Each site has multiple Lightweight Direct Access Protocol (LDAP) servers which are kept in sync using Oracle Unified Directory (OUD) replication.
- Site–1 has an administration server for the entire OIG deployment which spans both sites.

- Site–1 and Site–2 have independent Administration servers for the local OAM deployment.

About the Entry Points in Multi-Data Center Deployment

Access and Identity will have different entry points as described in the this guidePreparing the Load Balancer and Firewalls for an Enterprise Deployment. The reason for having two different entry points is that each entry point can be configured independently. For example:

- `login.example.com` will be configured to geographically distribute the requests amongst all active Oracle Access Management sites.
- `prov.example.com` will be configured to only send traffic to the active site in an active-passive deployment.

About the Databases in Multi-Data Center Deployment

Two different databases are required to support Oracle Identity and Access Management Suite because the changes are propagated between the sites in different ways (unless using the stretched cluster design for both Oracle Access Management (OAM) and Oracle Identity Governance (OIG).

The synchronization of OAM data is performed using proprietary OAM technology. This technology effectively unloads data from the primary site, transfers it to the secondary site, and applies it to the database on that site, using SQL commands. The databases on the primary and secondary sites are independent and are both open to read and write.

The synchronicity requirements and data types used by the different OIG components limit the possible approaches for the OIG database in a multi-data center deployment. This document addresses only a solution where the Oracle Fusion Middleware database used for OIG uses Data Guard to synchronize an active database in Site–1 with a passive database in Site–2. Although other approaches may work, they have not been tested and certified by Oracle and are out of the scope of this document.

In this configuration, we assume that both sites where OIG is deployed, access the same database (as well as the same schemas within that database), and the database is set up in a Data Guard configuration. Data Guard provides a comprehensive data protection solution for the database. It consists of a standby Site 1 at geographically different location than the production site. The standby database is normally in passive mode; it is started when the production site (called production, from the database activity point of view) is not available.

The Oracle databases configured in each site are in an Oracle Real Application Cluster (RAC). Oracle RAC enables an Oracle database to run across a cluster of servers in the same data center; providing fault tolerance, performance, and scalability with no application changes necessary.

In order to facilitate the smooth transformation of database transactions from one site to the other, a role-based database service is created on both the primary and standby database sites. A role-based service is only available when the database is running in the primary role, that is, when the database is open read/write. When a standby database becomes a primary database, the service is automatically enabled on that side.

By configuring the WebLogic data sources to use this role-based service and making the data sources aware of both sites, WebLogic reconfiguration is not required, when the primary database moves between sites

About the Directory Tier in Multi-Data Center Deployment

The Multi-Data Center deployment for Oracle Identity Governance (OIG) has been tested using Oracle Unified directory (OUD).

OUD is a loosely coupled deployment. Data is replicated between the OUD instances using OUD replication. The second site is just an extension of that principle with the remote OUD instances becoming part of the OUD replication configuration.

About the Load Balancers in Multi-Data Center Deployment

The global load balancer (GLBR) is a load balancer configured to be accessible as an address by users of all of the sites and external locations. The device provides a virtual server which is mapped to a DNS name that is accessible to any client regardless of the site they will be connecting to. The GLBR directs traffic to either Site based on configured criteria and rules. For example, the criteria can be based on the client’s IP. These criteria and rules should be used to create a Persistence Profile which allows the GLBR to map users to the same site on initial and subsequent requests. The GLBR maintains a pool, which consists of the addresses of all the local load balancers. In the event of failure of one of the sites, users are automatically redirected to the surviving active site.

At each site, Local Load Balancer (LBR) receives the request from GLBR and directs requests to the appropriate HTTP server. In either case, the LBR is configured with a persistence method such as Active Insert of a cookie in order to maintain affinity and ensure that clients are directed appropriately. To eliminate undesired routings and costly re-hydrations, the GLBR is also configured with specific rules that route callbacks only to the LBR that is local to the servers that generated them. This is useful also for internal consumers of Oracle Identity and Access Management services.

The GLBR rules can be summarized as follows:

- If requests come from Site–1 (callbacks from the Oracle Identity and Access Management servers in Site–1 or endpoint invocations from consumers in Site–1), the GLBR routes to the local load balancer (LBR) in Site–1.
- If requests come from Site–2 (callbacks from the Oracle Identity and Access Management servers in Site–2 or endpoint invocations from consumers in Site–2) the GLBR routes to the LBR in Site–2.
- If requests come from any other address (client invocations) the GLBR load balances the connections to both LBRs.
- Client requests may use the GLBR to direct requests to the nearest site geographically.
- Additional routing rules may be defined in the GLBR to route specific clients to specific sites (for example, the two sites may provide difference response time based on the hardware resources in each case).

Load balancers from any vendor are supported as long as the load balancer meets the requirements listed in Characteristics of the Oracle Access Management Design. The global load balancer should allow rules based on the originating server’s IPs (an example is provided for F5 Networks).

Shared Storage Versus Database for Transaction Logs and Persistent stores

The topology illustrated in About the Multi-Data Center Deployment Topology was tested using database-based persistent stores for Oracle WebLogic Server transactions logs and Oracle WebLogic Server JMS persistent stores.

Storing transaction logs and persistent stores in the database provides the replication and high availability benefits inherent from the underlying database system. With JMS, TLOG, and OIM/SOA data in a Data Guard database, cross-site synchronization is simplified and the need for a shared storage sub-system such as a NAS or a SAN is alleviated in the middle tier (they still apply for the Administration Server’s failover). Using TLOGs and JMS in the database has a penalty, however, on the system’s performance.

As of Oracle Fusion Middleware 11*g*, the retry logic in JMS JDBC persistent stores that takes care of failures in the database is limited to a single retry. If a failure occurs in the second attempt, an exception is propagated up the call stack and a manual restart of the server is required to recover the messages associated with the failed transaction (the server will go into FAILED state due to the persistent store failure). To overcome this faliure, it is recommended to use Test Connections on Reserve for the pertaining data sources and also configure in-place restart for the pertaining JMS Server and persistent stores. Refer to Appendix B for details on configuring in-place restart.

About the Characteristics of a Multi-Data Center Deployment

It is important to understand the characteristics of a multi-data center deployment and how to treat the multi-site implementations of Identity and Access differently, by separating the domains.

- Oracle Access Management (OAM) can use the proprietary multi-site technologies built into the product.
- Oracle Identity Manager (OIM) can use either the active-passive approach where network latency is high, or if that network latency is low, then be active-active as well. If Governance is running active-passive, the OIM components in Site–2 are shutdown until required.
- The Global Load Balancer (GLBR) is configured to send traffic only to the active site.
- Each OAM domain has a distinct entry point for administrative functions.
- Failover of the OIM Administration server can be accomplished using disk-based replication of the *IGD_ASERVER_HOME* directory and a virtual IP address (VIP), which can be moved between sites.
- Failure of the OAM Administration server is handled within the site using standard EDG techniques.

Characteristics of a Multi-Data Center Deployment

Availability (Web Tier):

The Oracle HTTP Server (OHS) configuration is based on a fixed list of servers in each site (instead of the dynamic list, provided by the OHS plug-in and used in typical single-location deployments). This configuration is done to eliminate undesired routing from one site to another, however, this configuration has the disadvantage of slower reaction times to failures in the Oracle WebLogic Server.

The following sections describe the characteristics of the OAM design and OIM design:

- Characteristics of the Oracle Access Management Design
  Review the characteristics of an Oracle Access Management (OAM) design.
- Characteristics of the Oracle Identity Governance Design
  Review the characteristics of an Oracle Identity Governance (OIG) design.

Characteristics of the Oracle Access Management Design

Review the characteristics of an Oracle Access Management (OAM) design.

- Availability

  In an OAM multi-data center deployment, there is little impact on operations, as each site is independent. In the OAM multi-data center deployment, one site is nominated as primary; if that site fails, the primary role has to be passed on to Site–2. This affects the creation of policies only; runtime is not be affected. However, design considerations should be made for runtime.

  If a request is authorized at Site 1, and Site 1 becomes unavailable, you have the option to force the user to re-authenticate at Site–2 to accept the authentication that has already occurred.

- Administration

  In a multi-data center active-active deployment, each site is independent of the other. The OAM replication mechanism takes care of the replication of OAM data, but the WebLogic or application configuration needs to be performed at each site independently. OAM does not use runtime artifacts (with the exception of the authentication cookie), so that the process is simplified,. However, having multiple independent configurations increases the administration overhead.

- Performance

  If the appropriate load balancing and traffic restrictions are configured as described in the Load Balancing section, the performance of OAM across sites should be similar to that of a cluster with the same number of servers residing in one single site.

- Load Balancing

  In an OAM deployment, the load balancer virtual hosts are as described in the guide with the following differences:

  - `login.example.com`: This virtual server is configured both locally and at the global load balancer level with location affinity.
  - `iadadmin.example.com`: This virtual server is unique to each site, that is, there is `iadadminsite1.example.com` and `iadadminsite2.example.com`.
  - `oam.example.com`: This virtual server is an additional load balancer entry point which is resolvable in each site and routes requests to the OAM Proxy port on the OAM Managed servers, for example 5575. This load balancer entry point is also configured at the Global Load Balancer (GLBR) level and distribute requests to each of the OAM Managed servers in both Multi-datacenter domains with location affinity. The advantage of configuring it at the GLBR level is that if the managed servers in Site–1 become unavailable but the web tier is available then authentications can still happen. The down side to this approach is that it will generate a lot of cross-site traffic. As each data center is high availability (HA) in its own right, it is unlikely that just the two hosts are effected by an outage, however, if both managed servers are down, all traffic is redirected to the second site. To redirect the traffic, you need to configure the

`login.example.com` monitoring service to monitor not just the web servers but also the availability of the OAM service.

> **Note:**
>
> As the Oracle Access Protocol (OAP) requests are being handled by the load balancer, a delay can result whilst the load balancer detects that an OAM server is not available.

Characteristics of the Oracle Identity Governance Design

Review the characteristics of an Oracle Identity Governance (OIG) design.

- Availability

  The database connection failover behavior and the JMS and RMI failover behaviors are similar to those that take place in a standard enterprise deployment topology. At all times, there is just one single *`CLUSTER_MASTER`* server, among all the available servers in the multi-data center active-active deployment, which is able to perform automatic recovery. Instances can be recovered equally from Site–1 and Site–2, if a failure occurs on the partner site as follows:

  - From Site–1, when Site–2 is up if the *`CLUSTER MASTER`* resides in Site–1.
  - From Site–2, when Site–1 is up if the *`CLUSTER MASTER`* resides in Site–2.
  - From Site–1, when Site–2 is down.
  - From Site–2, when Site–1 is down.

  If a failure occurs in Site–1 that affects all of the middle tiers, recovery of the administration Server is required to resume the Oracle Enterprise Manager Fusion Middleware Control and the Oracle WebLogic Server Administration Console.

  The servers that are remote to the Administration Server take longer to restart than in a regular Enterprise Deployment Topology because all communications with the Administration Server (for retrieving the domain configuration upon start), initial connection pool creation, and database access is affected by the latency across sites.

  From the perspective of Recovery Point Objective (RPO), transactions that are halted by a site failure can be resumed in the site, which is available, by manually starting the failed servers in that site. Automated server migration across sites is not recommended unless a database is used for JMS and TLOG persistence; otherwise a constant replica of the appropriate persistent stores needs to be set up between the sites. It is also unlikely (depending on the customer's infrastructure) that the virtual IPs used in one site are valid for migration to the other, as this usually requires additional intervention to enable a listen address initially available in Site–1 to Site–2 and vice versa. This intervention can be automated in pre-migration scripts, but in general, the Recovery time Objective (RTO) increases compared to a standard automated server migration (taking place in the scope of single data center).

- Performance

  If the appropriate load balancing and traffic restrictions are configured as described in the Load Balancing section, the performance of a stretched cluster with low latency across sites should be similar to that of a cluster with the same number of servers residing in one single site. The configuration steps provided in the Load Balancing section are intended to constrain the traffic inside each site for the most common and normal operations. This isolation, however, is non-deterministic (for example, there is room for failover scenarios where a JMS invocation could take place across the two sites), which implies that most of the traffic takes place between the OIG Servers and the database. This is the key to the performance of the multi-data center running in an Active-Active scenario.

If the sites are separated by a high latency network, then OIG should be run active-passive to avoid significant performance degradation.

- Administration

  In a multi-data center active-active deployment, the Oracle WebLogic Server Infrastructure is responsible for copying configuration changes to all the different domain directories used in the domain. The Coherence cluster configured is responsible for updating all the servers in the cluster when composites or metadata are updated. Except for the replication requirement for runtime artifacts across file systems, a multi-data center active-active deployment is administrated like a standard cluster, which makes its administration overhead very low.

Part II

Preparing for an Enterprise Deployment

It is important to understand the tasks that need to be performed to prepare for an enterprise deployment.

This part of the enterprise deployment guide contains the following topics.

- Using the Enterprise Deployment Workbook
  The Enterprise Deployment workbook enables you to plan an enterprise deployment for your organization.
- Procuring Resources for an Enterprise Deployment
  It is essential to procure the required hardware, software, and network settings before you configure the Oracle Identity and Access Management reference topology.
- Preparing the Load Balancer and Firewalls for an Enterprise Deployment
  It is important to understand how to configure the hardware load balancer and ports that must be opened on the firewalls for an enterprise deployment.
- Preparing the File System for an Enterprise Deployment
  Preparing the file system for an enterprise deployment involves understanding the requirements for local and shared storage, as well as the terminology that is used to reference important directories and file locations during the installation and configuration of the enterprise topology.
- Preparing the Operating System for an Oracle Identity and Access Management Deployment
  Preparing the operating system consists of performing all of the previous preparatory steps. Once completed, the environment will have the same structure as a traditional server deployment.
- Preparing the Host Computers for an Enterprise Deployment
  It is important to perform a set of tasks on each computer or server before you configure the enterprise deployment topology. This involves verifying the minimum hardware and operating system requirements for each host, configuring operating system users and groups, enabling Unicode support, mounting the required shared storage systems to the host and enabling the required virtual IP addresses on each host.
- Preparing the Database for an Enterprise Deployment
  Preparing the database for an enterprise deployment involves ensuring that the database meets specific requirements, creating database services, using SecureFiles for large objects in the database, and creating database backup strategies.

5 Using the Enterprise Deployment Workbook

The Enterprise Deployment workbook enables you to plan an enterprise deployment for your organization.

This chapter provides an introduction to the Enterprise Deployment workbook, use cases, and information on who should use the Enterprise Deployment workbook.

- Introduction to the Enterprise Deployment Workbook
  The Enterprise Deployment workbook is a spreadsheet that is used by architects, system engineers, database administrators, and others to plan and record all the details for an environment installation (such as server names, URLs, port numbers, installation paths, and other resources).
- Typical Use Case for Using the Workbook
  It is important to understand the roles and tasks involved in a typical use case of the Enterprise Deployment workbook.
- Who Should Use the Enterprise Deployment Workbook?
  The details of the Enterprise Deployment workbook are filled in by the individual or a team that is responsible for planning, procuring, or setting up each category of resources.
- Using the Oracle Identity and Access Management Enterprise Deployment Workbook
  Locating and understanding the Oracle Identity and Access Management Enterprise Deployment Workbook enables you to use it efficiently.

Introduction to the Enterprise Deployment Workbook

The Enterprise Deployment workbook is a spreadsheet that is used by architects, system engineers, database administrators, and others to plan and record all the details for an environment installation (such as server names, URLs, port numbers, installation paths, and other resources).

The Enterprise Deployment workbook serves as a single document that you can use to track input variables for the entire process, allowing for:

- Separation of tasks between architects, system engineers, database administrators, and other key organizational roles.
- Comprehensive planning before the implementation.
- Validation of planned decisions before the actual implementation.
- Consistency during implementation.
- A record of the environment for future use.

Typical Use Case for Using the Workbook

It is important to understand the roles and tasks involved in a typical use case of the Enterprise Deployment workbook.

A typical use case for the Enterprise Deployment workbook involves the following roles and tasks, in preparation for an Oracle Fusion Middleware Enterprise Deployment:

- Architects read through the first five chapters of this guide, and fill in the corresponding sections of the workbook.
- The workbook is validated by other architects and system engineers.
- The architect uses the validated workbook to initiate network and system change requests with the system engineering departments.
- The Administrators and System Integrators who install and configure the software refer to the workbook and the subsequent chapters of this guide to perform the installation and configuration tasks.

Who Should Use the Enterprise Deployment Workbook?

The details of the Enterprise Deployment workbook are filled in by the individual or a team that is responsible for planning, procuring, or setting up each category of resources.

The information in the Enterprise Deployment workbook is divided into categories. Depending on the structure of your organization and roles that are defined for your team, you can assign specific individuals in your organization to fill in the details of the workbook. Similarly, the information in each category can be assigned to the individual or team that is responsible for planning, procuring, or setting up each category of resources.

For example, the workbook can be filled in, reviewed, and used by people in your organization that fill the following roles:

- Information Technology (IT) Director
- Architect
- System Administrator
- Network Engineer
- Database Administrator

Using the Oracle Identity and Access Management Enterprise Deployment Workbook

Locating and understanding the Oracle Identity and Access Management Enterprise Deployment Workbook enables you to use it efficiently.

The following sections provide an introduction to the location and contents of the Oracle Identity and Access Management Enterprise Deployment Workbook:

- Locating the Oracle Identity and Access Management Enterprise Deployment Workbook
- Understanding the Contents of the Oracle Identity and Access Management Enterprise Deployment Workbook

Locating the Oracle Identity and Access Management Enterprise Deployment Workbook

The Oracle Identity and Access Management Enterprise Deployment Workbook is available as a Microsoft Excel Spreadsheet in the Oracle Fusion Middleware documentation library. It is available as a link on the Install, Patch, and Upgrade page of the library.

Understanding the Contents of the Oracle Identity and Access Management Enterprise Deployment Workbook

The following sections describe the contents of the Oracle Identity and Access Management Enterprise Deployment Workbook. The workbook is divided into tabs, each containing a set of related variables and values you will need to install and configure the Oracle Identity and Access Management Enterprise Deployment topologies:

- Using the Start Tab
- Using the Hardware - Host Computers Tab
- Using the Network - Virtual Hosts & Ports Tab
- Using the Load Balancer Tab
- Using the Storage - Directory Variables Tab
- Using the Database - Connection Details Tab
- Using the LDAP - Users and Groups Tab
- Using the Operating System Tab

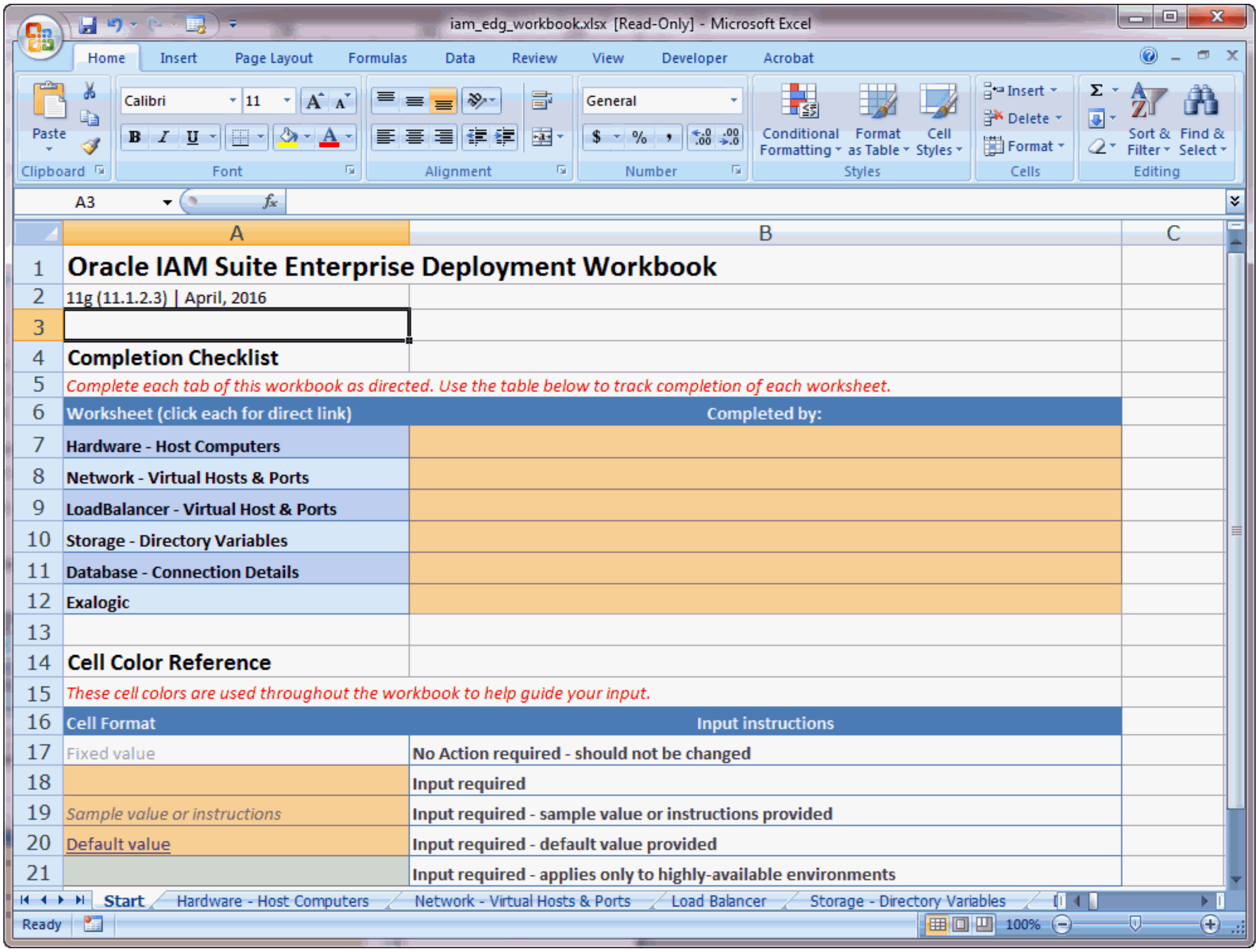
Using the Start Tab

The Start tab of the Enterprise Deployment Workbook serves as a table of contents for the rest of the workbook. You can also use it to identify the people who will be completing the spreadsheet.

The Start tab also provides a key to identify the colors used to identify workbook fields that need values, as well as those that are provided for informational purposes.

Figure 5-1 shows the Start tab of the spreadsheet.

Figure 5-1 Start Tab of the Oracle Identity and Access Management Enterprise Deployment Workbook



	A	B	C
1	Oracle IAM Suite Enterprise Deployment Workbook		
2	11g (11.1.2.3) | April, 2016		
3			
4	Completion Checklist		
5	*Complete each tab of this workbook as directed. Use the table below to track completion of each worksheet.*		
6	Worksheet (click each for direct link)	Completed by:	
7	Hardware - Host Computers		
8	Network - Virtual Hosts & Ports		
9	LoadBalancer - Virtual Host & Ports		
10	Storage - Directory Variables		
11	Database - Connection Details		
12	Exalogic		
13			
14	Cell Color Reference		
15	*These cell colors are used throughout the workbook to help guide your input.*		
16	Cell Format	Input instructions	
17	Fixed value	No Action required - should not be changed	
18		Input required	
19	*Sample value or instructions*	Input required - sample value or instructions provided	
20	Default value	Input required - default value provided	
21		Input required - applies only to highly-available environments	

Using the Hardware - Host Computers Tab

The Hardware - Host Computers tab lists the host computers required to install and configure the Oracle Identity and Access Management Enterprise Deployment Topology.

The reference topologies described in About the Primary and Build-Your-Own Enterprise Deployment Topologies require a minimum of six host computers: two for the Web tier, two for the application tier, and two for the Oracle RAC database on the data tire.

A common deployment model typically uses 10 servers however. These being made up of: 2 for the Web Tier, 2 for the Access Components Application Tier, 2 for the Governance Components Application Tier, 2 For the LDAP servers and 2 for the RAC database servers. If you decide to expand the environment to include more systems, add a row for each additional host computer.

The **Abstract Host Name** is the name used throughout this guide to reference the host. For each row, procure a host computer, and enter the **Actual Host Name**.

For example, if a procedure in this guide references OAMHOST1, you can then replace the OAMHOST1 variable with the actual name provided on the **Hardware - Host Computers** tab of the workbook.

About Multi-Networked Host Computers

If you are deploying on a multi-networked host, the real host name may not be attached to the network on which you wish communication to occur. If the network you wish to use for communication is different from that attached to the **Real Host Name**, then you can override this by providing a different **Listen Address Host Name**, which is attached to the network you wish to use. Most platform deployments do not require a different **Listen Host Name**, however the majority of Exalogic Deployments do.

A typical example would be where the real host name is attached to the management network but network communication should happen through a client network or in the case of Exalogic the internal IPoIB network.

Using the Spreadsheet in a Consolidated Deployment

If you are using a consolidated deployment, where you have larger machines, then you can use the same host name for multiple entries in the spreadsheet.

For example, if you wish to deploy Access and Governance onto the same host then both OAMHOST1 and OIMHOST1 can be set to *iamserver1*, and both OAMHOST1 and OIMHOST2 can be set to *iamserver2*.

When you see OAMHOST1 or OIMHOST2 referenced in the guide, you'll know to replace them with the value of *iamserver1* or *iamserver2*.

Including Additional Host Details

For easy reference, Oracle also recommends that you include the IP address, Operating System (including the version), number of CPUs, and the amount of RAM for each host. This information can be useful during the installation, configuration, and maintenance of the enterprise deployment.

Using the Network - Virtual Hosts & Ports Tab

The Network - Virtual Hosts & Ports tab lists the virtual hosts that must be defined by your network administrator before you can install and configure the enterprise deployment topology.

The port numbers are important for several reasons. You must have quick reference to the port numbers so that you can access the management consoles; the firewalls must also be configured to allow network traffic through specific ports.

Each virtual host, virtual IP address, and each network port serves a distinct purpose in the deployment. See Preparing the Load Balancer and Firewalls for an Enterprise Deployment.

In the Network - Virtual Hosts table, review the items in the **Abstract Virtual Host or Virtual IP Name** column. These are the virtual host and virtual IP names that are used in the procedures in this guide. For each abstract name, enter the actual virtual host name that is defined by your network administrator. Whenever this guide references one of the abstract virtual host or virtual IP names, replace that value with the actual corresponding value in this table.

Similarly, in many cases, this guide assumes that you are using default port numbers for the components or products you install and configure. However, in reality, you are likely to use different port numbers. Use the Network - Port Numbers table to map the default port values to the actual values that are used in your specific installation.

Using the Load Balancer Tab

The Load Balancer tab lists the virtual hosts your network administrator must create on your hardware load balancer before you can install and configure the Oracle IAM enterprise deployment topology.

The ports you specify in this section are the ports on the load balancer. They need not be the same as the target ports you are directing traffic to.

Each virtual host, virtual IP address, and each network port serves a distinct purpose in the deployment.

The Virtual Hosts are separated out to provide maximum flexibility. It is however acceptable to combine the multiple virtual hosts of the same type.

In the **Load Balancer - Virtual Hosts** table, review the items in the **Abstract Virtual Host** or **Virtual IP Name** column. These are the virtual host and virtual IP names used in the procedures in this guide. For each abstract name, enter the actual virtual host name defined by your network administrator. Whenever this guide references one of the abstract virtual host or virtual IP names, replace that value with the actual corresponding value in this table.

Similarly, in many cases, this guide assumes you are using default port numbers for the components or products you install and configure. However, in reality, you will likely have to use different port numbers. Use the **Load Balancer - Port Numbers** table to map the default port values to the actual values used in your specific installation.

The Load Balancer Pool configuration combines information that you enter in this tab with information entered in the **Hardware** and **Network** tabs to provide a summary of how the load balancer pools should be configured.

Using the Storage - Directory Variables Tab

As part of preparing for an enterprise deployment, it is assumed you are using a standard directory structure, which is recommended for Oracle enterprise deployments.

In addition, procedures in this book reference specific directory locations. Within the procedures, each directory is assigned a consistent variable, which you should replace with the actual location of the directory in your installation.

For each of the directory locations listed on this tab, provide the actual directory path in your installation.

In addition, for the application tier, it is recommended that many of these standard directories be created on a shared storage device. For those directories, the table also provides fields so you can enter the name of the shared storage location and the mount point that is used when you mounted the shared location. See Preparing the File System for an Enterprise Deployment.

Using the Database - Connection Details Tab

When you are installing and configuring the enterprise deployment topology, you will often have to make connections to a highly available Oracle Real Application Clusters (RAC) database. In this guide, the procedures reference a set of variables that identify the information you will need to provide to connect to the database from tools, such as the Configuration Wizard and the Repository Creation Utility.

To be sure you have these values handy, use this tab to enter the actual values for these variables in your database installation.

An Oracle Identity and Access Management installation can use more than one database if desired. This is typically the case where you wish to use a Multi Data Center deployment. It is perfectly acceptable however, to use a single database.

If you are using a single database, you must still use a different RCU prefix for artefacts belonging to each separate domain Access and Governance.

Using the LDAP - Users and Groups Tab

When you are installing and configuring the enterprise deployment topology, you will need to create users and groups to be imported into the OUD server. These users and groups are required for populating the LDAP server for the various Oracle Access Manager and Oracle Identity Governance applications.

To be sure you have these values handy, use the LDAP - Users and Groups tab to enter the actual values for these variables in your LDAP installation and configuration.

Using the Operating System Tab

This tab is used to document the required operating system configurations. These include connectivity, virtual server details, and share requirements.

6 Procuring Resources for an Enterprise Deployment

It is essential to procure the required hardware, software, and network settings before you configure the Oracle Identity and Access Management reference topology.

This chapter provides information on how to reserve the required IP addresses and identify and obtain software downloads for an enterprise deployment.

- Hardware and Software Requirements for the Enterprise Deployment Topology
  It is important to understand the hardware load balancer requirements, host computer hardware requirements, and operating system requirements for the enterprise deployment topology.
- Reserving the Required IP Addresses for an Enterprise Deployment
  You have to obtain and reserve a set of IP addresses before you install and configure the enterprise topology. The set of IP addresses that need to be reserved are listed in this section.
- Identifying and Obtaining Software Distributions for an Enterprise Deployment
  Before you begin to install and configure the enterprise topology, you must obtain the software distributions that you need to implement the topology.

Hardware and Software Requirements for the Enterprise Deployment Topology

It is important to understand the hardware load balancer requirements, host computer hardware requirements, and operating system requirements for the enterprise deployment topology.

This section includes the following sections.

- Hardware Load Balancer Requirements
  The section lists the wanted features of the external load balancer.
- Host Computer Hardware Requirements
  This section provides information to help you procure host computers that are configured to support the enterprise deployment topologies.
- Operating System Requirements for an Enterprise Deployment Topology
  This section provides details about the operating system requirements.
- Virtual Server Requirements
- About Private Networks
- About Virtual Server Templates

Hardware Load Balancer Requirements

The section lists the wanted features of the external load balancer.

The enterprise topology uses an external load balancer. The features of the external load balancer are:

- Ability to load-balance traffic to a pool of real servers through a virtual host name: Clients access services by using the virtual host name (instead of using actual host names). The load balancer can then load balance requests to the servers in the pool.
- Port translation configuration should be possible so that incoming requests on the virtual host name and port are directed to a different port on the backend servers.
- Monitoring of ports on the servers in the pool to determine availability of a service.
- Ability to configure names and ports on your external load balancer. The virtual server names and ports must meet the following requirements:
  - The load balancer should allow configuration of multiple virtual servers. For each virtual server, the load balancer should allow configuration of traffic management on more than one port. For example, for Oracle HTTP Server in the web tier, the load balancer needs to be configured with a virtual server and ports for HTTP and HTTPS traffic.
  - The virtual server names must be associated with IP addresses and be part of your DNS. Clients must be able to access the external load balancer through the virtual server names.
- Ability to detect node failures and immediately stop routing traffic to the failed node.
- It is highly recommended that you configure the load balancer to be in fault-tolerant mode.
- It is highly recommended that you configure the load balancer virtual server to return immediately to the calling client when the backend services to which it forwards traffic are unavailable. This is preferred over the client disconnecting on its own after a timeout based on the TCP/IP settings on the client machine.
- Ability to maintain sticky connections to components. Examples of this include cookie-based persistence, IP-based persistence, and so on.
- The load balancer should be able to terminate SSL requests at the load balancer and forward traffic to the backend real servers by using the equivalent non-SSL protocol (for example, HTTPS to HTTP).
- SSL acceleration (this feature is recommended, but not required for the enterprise topology).

Host Computer Hardware Requirements

This section provides information to help you procure host computers that are configured to support the enterprise deployment topologies.

It includes the following topics.

- General Considerations for Enterprise Deployment Host Computers
  This section specifies the general considerations that are required for the enterprise deployment host computers.
- Reviewing the Oracle Fusion Middleware System Requirements
  This section provides reference to the system requirements information to help you ensure that the environment meets the necessary minimum requirements.

- Typical Memory, File Descriptors, and Processes Required for an Enterprise Deployment
  This section specifies the typical memory, number of file descriptors, and operating system processes and tasks details that are required for an enterprise deployment.
- Typical Disk Space Requirements for an Enterprise Deployment
  This section specifies the disk space that is typically required for this enterprise deployment.

General Considerations for Enterprise Deployment Host Computers

This section specifies the general considerations that are required for the enterprise deployment host computers.

Before you start the process of configuring an Oracle Fusion Middleware enterprise deployment, you must perform the appropriate capacity planning to determine the number of nodes, CPUs, and memory requirements for each node depending on the specific system's load as well as the throughput and response requirements. These requirements vary for each application or custom Oracle Identity and Access Management system being used.

The information in this chapter provides general guidelines and information that helps you determine the host computer requirements. It does not replace the need to perform capacity planning for your specific production environment.

> **Note:**
>
> As you obtain and reserve the host computers in this section, note the host names and system characteristics in the Enterprise Deployment workbook. You will use these addresses later when you enable the IP addresses on each host computer. See Using the Enterprise Deployment Workbook.

Reviewing the Oracle Fusion Middleware System Requirements

This section provides reference to the system requirements information to help you ensure that the environment meets the necessary minimum requirements.

Review the *Oracle Fusion Middleware System Requirements and Specifications* to ensure that your environment meets the minimum installation requirements for the products that you are installing.

The Requirements and Specifications document contains information about general Oracle Fusion Middleware hardware and software requirements, minimum disk space and memory requirements, database schema requirements, and the required operating system libraries and packages.

It also provides some general guidelines for estimating the memory requirements for your Oracle Fusion Middleware deployment.

Typical Memory, File Descriptors, and Processes Required for an Enterprise Deployment

This section specifies the typical memory, number of file descriptors, and operating system processes and tasks details that are required for an enterprise deployment.

The following table summarizes the memory, file descriptors, and processes required for the Administration Server and each of the Managed Servers computers in a typical Oracle Identity and Access Management enterprise deployment. These values are provided as an example only, but they can be used to estimate the minimum amount of memory required for an initial enterprise deployment.

The example in this topic reflects the minimum requirements for configuring the Managed Servers and other services required on OAMHOST1, as depicted in the reference topologies.

When you procure systems, use the information in the **Approximate Top Memory** column as a guide when determining the minimum physical memory that each host computer should have available.

After you procure the host computer hardware and verify the operating system requirements, review the software configuration to be sure that the operating system settings are configured to accommodate the number of open files listed in the **File Descriptors** column and the number processes listed in the **Operating System Processes and Tasks** column.

See Setting the Open File Limit and Number of Processes Settings on UNIX Systems.

Managed Server, Utility, or Service	Approximate Top Memory	Number of File Descriptors	Operating System Processes and Tasks
Access Administration Server	3.5 GB	2300	180
Governance Administration Server	3.5 GB	2100	100
WLS_SOA	2.0 GB	1400	210
WLS_OIM	8 GB	1400	190
WLS_OAM	1.0 GB	2000	170
WLS_AMA	2.0 GB	1700	160
WLS_WSM	3.0 GB	200	130
WLST (connection to the Node Manager)	1.5 GB	910	20
Configuration Wizard	1.5 GB	700	20
Node Manager	268 MB	300	20
Node Manager (per domain)	1.0 GB	720	15
TOTAL	22.0 GB*	14430	805

* Approximate total, with consideration for Operating System and other additional memory requirements.

Typical Disk Space Requirements for an Enterprise Deployment

This section specifies the disk space that is typically required for this enterprise deployment.

For the latest disk space requirements for the Oracle Fusion Middleware 12c (12.2.1.4.0) products, including the Oracle Identity and Access Management products, review the *Oracle Fusion Middleware System Requirements and Specifications*.

In addition, the following table summarizes the disk space that is typically required for an Oracle Identity and Access Management enterprise deployment.

Use the this information and the information in Preparing the File System for an Enterprise Deployment to determine the disk space requirements required for your deployment.

Server	Disk
Database	nXm n = number of disks, at least 4 (striped as one disk) m = size of the disk (minimum of 30 GB)
WEBHOST*n*	10 GB
OAMHOST*n*	10 GB*
OIMHOST*n*	10 GB*
LDAPHOST*n*	10 GB*

* For a shared storage Oracle home configuration, two installations suffice by making a total of 20 GB.

** For Portal servers running Elasticsearch, the baseline `ES_HOME` directory without index data will require approximately 60 MB. Additional storage for logs and search indexes would increase based on quantity and textual size of the documents and portals crawled by the Portal and indexed in Elasticsearch.

Operating System Requirements for an Enterprise Deployment Topology

This section provides details about the operating system requirements.

The Oracle Fusion Middleware software products and components that are described in this guide are certified on various operating systems and platforms, which are listed in *Oracle Fusion Middleware System Requirements and Specifications*.

> **Note:**
>
> This guide focuses on the implementation of the enterprise deployment reference topology on Oracle Linux systems.
>
> The topology can be implemented on any certified, supported operating system, but the examples in this guide typically show the commands and configuration steps as they should be performed by using the bash shell on Oracle Linux.

Virtual Server Requirements

If you are deploying onto a Virtual deployment, you will need to create the following virtual servers to host a typical Oracle Identity and Access Management Enterprise Deployment.

> **Note:**
>
> As you obtain and reserve the host computers in this section, note the host names and system characteristics in the Enterprise Deployment Workbook. Use these addresses later when you enable the IP addresses on each host computer. For more information, see Using the Enterprise Deployment Workbook .

- Virtual Servers Required for IAM
- About Distribution Groups

Virtual Servers Required for IAM

When you deploy the Oracle Identity and Access Management software as part of a virtual configuration, you must be sure to configure the vServers.

Table 6-1 vServer Information

Name	vServerType	Virtual Networks	Host Name	Distribution Group
WEBHOST1	LARGE	IPoIB-EDG[1] EoIB-client[2] IPoIB-Storage[3]	WEBHOST1 WEBHOST1-ext WEBHOST1-stor	IAM_WEB
WEBHOST2	LARGE	IPoIB-EDG EoIB-client IPoIB-Storage	WEBHOST2 WEBHOST2-ext WEBHOST2-stor	IAM_WEB
OAMHOST1	EXTRA_LARGE	IPoIB-EDG EoIB-client IPoIB-Storage	OAMHOST1 OAMHOST1-ext OAMHOST1-stor	IAM_IAD
OAMHOST2	EXTRA_LARGE	IPoIB-EDG EoIB-client IPoIB-Storage	OAMHOST2 OAMHOST2-ext OAMHOST2-stor	IAM_IAD
OIMHOST1	EXTRA_LARGE	IPoIB-EDG EoIB-client IPoIB-Storage	OIMHOST1 OIMHOST1-ext OIMHOST1-stor	IAM_IAG
OIMHOST2	EXTRA_LARGE	IPoIB-EDG EoIB-client IPoIB-Storage	OIMHOST2 OIMHOST2-ext OIMHOST2-stor	IAM_IAG
LDAPHOST1	EXTRA_LARGE	IPoIB-EDG IPoIB-Storage	LDAPHOST1 LDAPHOST1-stor	IAM_LDAP
LDAPHOST2	EXTRA_LARGE	IPoIB-EDG IPoIB-Storage	LDAPHOST2 LDAPHOST2-stor	IAM_LDAP

[1] IPoIB-EDG is the internal IPoIB network used for inter vServer communication. This is only required if you plan to use this network. If you plan on putting everything on the EoIB network, this is not required.

[2] EoIB-client is the Client Access Network which connects to the corporate ethernet

[3] IPoIB-Storage is the internal network that vServers use to communicate with the ZFS storage appliance.

About Distribution Groups

Distribution groups are used to ensure that the same application running in multiple virtual servers do not all run on the same physical host. By preventing different vServers of the same type running on the same physical server, you prevent the failure of the underlying physical server from taking out the complete system.

For an Oracle Fusion Middleware Enterprise Deployment you need to the following Distribution Groups:

- EDG_OAM: Prevents two IAMAccessDomain Servers from running on the same physical server
- EDG_OIM: Prevents two IAMGovernanceDomain Servers from running on the same physical server
- EDG_LDAP: Prevents two LDAP servers running on the same physical server.

About Private Networks

If you are going to keep interapp communication on the internal network, you must create a private VLAN.

About Virtual Server Templates

The following are the typical virtual server templates. You can customize these values depending on the results of your capacity planning.

Table 6-2 Virtual Server Templates

Type	Description	Memory (GB)	Number of Virtual CPUs
VERY_LARGE	Large Memory Intensive Applications	20	6
EXTRA_LARGE	CPU Intensive Applications	16	6
LARGE	Average Intensity Applications	8	2
SMALL	Low intensity applications	4	1

Reserving the Required IP Addresses for an Enterprise Deployment

You have to obtain and reserve a set of IP addresses before you install and configure the enterprise topology. The set of IP addresses that need to be reserved are listed in this section.

Before you begin installing and configuring the enterprise topology, you must obtain and reserve a set of IP addresses:

- Physical IP (IP) addresses for each of the host computers that you have procured for the topology
- A virtual IP (VIP) address for the Administration Server
- Additional VIP addresses for each Managed Server that is configured for Whole Server Migration

  For Fusion Middleware 12*c* products that support Automatic Service Migration, VIPs for the Managed Servers are typically not necessary.
- A unique virtual host name to be mapped to each VIP.

You can then work with your network administrator to be sure that these required VIPs are defined in your DNS server. Alternatively, for non-production environments, you can use the /etc/hosts file to define these virtual hosts.

For more information, see the following topics.

- What is a Virtual IP (VIP) Address?
  This section defines the virtual IP address and specifies its purpose.
- Why Use Virtual Host Names and Virtual IP Addresses?
  For an enterprise deployment, in particular, it is important that a set of VIPs--and the virtual host names to which they are mapped--are reserved and enabled on the corporate network.
- Physical and Virtual IP Addresses Required by the Enterprise Topology
  This section describes the physical IP (IP) and virtual IP (VIP) addresses that are required for the Administration Server and each of the Managed Servers in a typical Oracle Identity and Access Management enterprise deployment topology.

What is a Virtual IP (VIP) Address?

This section defines the virtual IP address and specifies its purpose.

A virtual IP address is an unused IP Address that belongs to the same subnet as the host's primary IP address. It is assigned to a host manually. If a host computer fails, the virtual address can be assigned to a new host in the topology. For the purposes of this guide, *virtual* IP addresses are referenced, which can be reassigned from one host to another, and *physical* IP addresses are referenced, which are assigned permanently to hardware host computer.

Why Use Virtual Host Names and Virtual IP Addresses?

For an enterprise deployment, in particular, it is important that a set of VIPs--and the virtual host names to which they are mapped--are reserved and enabled on the corporate network.

Alternatively, host names can be resolved through appropriate `/etc/hosts` file propagated through the different nodes.

In the event of the failure of the host computer where the IP address is assigned, the IP address can be assigned to another host in the same subnet, so that the new host can take responsibility for running the Managed Servers that are assigned to it.

The reassignment of virtual IP address for the Administration Server must be performed manually, but the reassignment of virtual IP addresses for Managed Servers can be performed automatically by using the Whole Server Migration feature of Oracle WebLogic Server.

Whether you should use Whole Server Migration or not depends upon the products that you are deploying and whether they support Automatic Service Migration.

Physical and Virtual IP Addresses Required by the Enterprise Topology

This section describes the physical IP (IP) and virtual IP (VIP) addresses that are required for the Administration Server and each of the Managed Servers in a typical Oracle Identity and Access Management enterprise deployment topology.

Before you begin to install and configure the enterprise deployment, reserve a set of host names and IP addresses that correspond to the VIPs in Table 6-3.

You can assign any unique host name to the VIPs, but in this guide, each VIP is referenced by using the suggested host names in the table.

Note:

As you obtain and reserve the IP addresses and their corresponding virtual host names in this section, note the values of the IP addresses and host names in the Enterprise Deployment workbook. You will use these addresses later when you enable the IP addresses on each host computer. See Using the Enterprise Deployment Workbook .

Table 6-3 Summary of the Virtual IP Addresses Required for the Enterprise Deployment

Virtual IP	VIP Maps to...	Description
VIP1	IADADMINVHN	IADADMINVHN is the virtual host name used as the listen address for the Administration Server used by the IAMAccessDomain and fails over with manual failover of the Administration Server. It is enabled on the node where the Administration Server process is running.
VIP2	IGDADMINVHN	IGDADMINVHN is the virtual host name used as the listen address for the Administration Server used by the IAMGovernacneDomain and fails over with manual failover of the Administration Server. It is enabled on the node where the Administration Server process is running.

If you are planning on using wholeserver migration rather than service migration for Oracle Identity Governance, you need the following additional VIPS:

Table 6-4 Summary of the Virtual IP Addresses Required for Whole Server Migration

Virtual IP	VIP Maps to...	Description
VIP1	OIMHOSTxVHN1	OIMHOSTxVHN1 is used by the WLS_OIM managed servers.
VIP2	OIMHOSTxVHN2	OIMHOSTxVHN2 is used by the WLS_SOA managed servers.

Table 6-4 (Cont.) Summary of the Virtual IP Addresses Required for Whole Server Migration

Virtual IP	VIP Maps to...	Description
VIP3	OIMHOSTxVHN3	OIMHOSTxVHN3 is used by the WLS_WSM managed servers.

In the above examples, `x` denotes the host name. For example, `OIMHOST1VHN1`.

Identifying and Obtaining Software Distributions for an Enterprise Deployment

Before you begin to install and configure the enterprise topology, you must obtain the software distributions that you need to implement the topology.

The following table lists the distributions used in this guide.

For general information about how to obtain Oracle Fusion Middleware software, see Obtaining Product Distributions in *Planning an Installation of Oracle Fusion Middleware*.

For more specific information about locating and downloading specific Oracle Fusion Middleware products, see the *Oracle Fusion Middleware Download, Installation, and Configuration Readme Files* on OTN.

> **Note:**
>
> The information in this guide is meant to complement the information contained in the Oracle Fusion Middleware certification matrixes. If there is a conflict of information between this guide and the certification matrixes, then the information in the certification matrixes must be considered the correct version, as they are frequently updated.

Distribution	Installer File Name	Description
Oracle Fusion Middleware 12c (12.2.1.4.0) Infrastructure	`fmw_12.2.1.4.0_infrastructure.jar`	Download this distribution to install the Oracle Fusion Middleware Infrastructure, which includes Oracle WebLogic Server and Java Required Files software required for Oracle Fusion Middleware products. This distribution also installs the Repository Creation Utility (RCU), which in previous Oracle Fusion Middleware releases was packaged in its own distribution.
Oracle HTTP Server 12c (12.2.1.4.0)	`fmw_12.2.1.4.0_ohs_linux64.bin`	Download this distribution to install the Oracle HTTP Server software on the Web Tier.
Oracle Unified Directory 12c (12.2.1.4.0)	`fmw_12.2.1.4.0_oud.jar`	Download this distribution to install the Oracle Unified Directory software.
Oracle Internet Directory 12c (12.2.1.4.0)	`fmw_12.2.1.4.0_oid_linux64.bin`	Download this distribution to install the Oracle Internet Directory software.

Distribution	Installer File Name	Description
Oracle Identity and Access Management 12c (12.2.1.4.0)	`fmw_12.2.1.4.0_idm.jar`	Download this distribution to install the Oracle Identity and Access Management software.
Oracle SOA Suite 12c (12.2.1.4.0)	`fmw_12.2.1.4.0_soa.jar`	Download this distribution to install the Oracle SOA Suite software.
Oracle Internet Directory Connector (12.2.1.3+)	`oid-12.2.1.3.0.zip`	Download this distribution to integrate with Oracle Internet Directory and Oracle Unified Directory.

> **Note:**
>
> Some of the functionality in this document requires that you apply Bundle Patch 2 (BP2) or later.

7 Preparing the Load Balancer and Firewalls for an Enterprise Deployment

It is important to understand how to configure the hardware load balancer and ports that must be opened on the firewalls for an enterprise deployment.

- Configuring Virtual Hosts on the Hardware Load Balancer
  The hardware load balancer configuration facilitates to recognize and route requests to several virtual servers and associated ports for different types of network traffic and monitoring.
- Configuring Global Load Balancers
  As indicated in the previous sections, the Global Load Balancer (GLBR) is responsible for performing smart routing of requests between multiple Local Load Balancers.
- Configuring the Firewalls and Ports for an Enterprise Deployment
  As an administrator, it is important that you become familiar with the port numbers that are used by various Oracle Fusion Middleware products and services. This ensures that the same port number is not used by two services on the same host, and that the proper ports are open on the firewalls in the enterprise topology.

Configuring Virtual Hosts on the Hardware Load Balancer

The hardware load balancer configuration facilitates to recognize and route requests to several virtual servers and associated ports for different types of network traffic and monitoring.

The following topics explain how to configure the hardware load balancer, provide a summary of the virtual servers that are required, and provide additional instructions for these virtual servers:

- Overview of the Hardware Load Balancer Configuration
- Typical Procedure for Configuring the Hardware Load Balancer
- Load Balancer Health Monitoring
- Summary of the Virtual Servers Required for an Enterprise Deployment
- Summary of the Virtual Servers Required for an Oracle Identity and Access Management Deployment

Overview of the Hardware Load Balancer Configuration

As shown in the topology diagrams, you must configure the hardware load balancer to recognize and route requests to several virtual servers and associated ports for different types of network traffic and monitoring.

In the context of a load-balancing device, a virtual server is a construct that allows multiple physical servers to appear as one for load-balancing purposes. It is typically represented by an IP address and a service, and it is used to distribute incoming client requests to the servers in the server pool.

The virtual servers should be configured to direct traffic to the appropriate host computers and ports for the various services that are available in the enterprise deployment.

In addition, you should configure the load balancer to monitor the host computers and ports for availability so that the traffic to a particular server is stopped as soon as possible when a service is down. This ensures that incoming traffic on a given virtual host is not directed to an unavailable service in the other tiers.

Note that after you configure the load balancer, you can later configure the web server instances in the web tier to recognize a set of virtual hosts that use the same names as the virtual servers that you defined for the load balancer. For each request coming from the hardware load balancer, the web server can then route the request appropriately, based on the server name included in the header of the request. See Configuring Oracle HTTP Server for Administration and Oracle Web Services Manager.

Typical Procedure for Configuring the Hardware Load Balancer

The following procedure outlines the typical steps for configuring a hardware load balancer for an enterprise deployment.

Note that the actual procedures for configuring a specific load balancer will differ, depending on the specific type of load balancer. There may also be some differences depending on the type of protocol that is being load balanced. For example, TCP virtual servers and HTTP virtual servers use different types of monitors for their pools. Refer to the vendor-supplied documentation for actual steps.

1. Create a pool of servers. This pool contains a list of servers and the ports that are included in the load-balancing definition.

   For example, for load balancing between the web hosts, create a pool of servers that would direct requests to hosts WEBHOST1 and WEBHOST2 on port 7777.

2. Create rules to determine whether a given host and service is available and assign it to the pool of servers that are described in Step 1.

3. Create the required virtual servers on the load balancer for the addresses and ports that receive requests for the applications.

   For a complete list of the virtual servers required for the enterprise deployment, see Summary of the Virtual Servers Required for an Enterprise Deployment.

   When you define each virtual server on the load balancer, consider the following:

   a. If your load balancer supports it, specify whether the virtual server is available internally, externally, or both. Ensure that internal addresses are only resolvable from inside the network.

   b. Configure SSL Termination, if applicable, for the virtual server.

   c. Assign the pool of servers created in Step 1 to the virtual server.

Load Balancer Health Monitoring

The load balancer must be configured to check that the services in the Load Balancer Pool are available. Failure to do so will result in requests being sent to hosts where the service is not running.

The following table shows examples of how to determine whether a service is available:

Table 7-1 Examples Showing How to Determine Whether a Service is Available

Service	Monitor Type	Monitor Mechanism
OUD	ldap	ldapbind to cn=oudadmin
OHS	http	check for GET /\r\n

Summary of the Virtual Servers Required for an Enterprise Deployment

This topic provides details of the virtual servers that are required for an enterprise deployment.

The following table provides a list of the virtual servers that you must define on the hardware load balancer for the Oracle Identity and Access Management enterprise topology:

Virtual Host	Server Pool	Protocol	SSL Termination?	Other Required Configuration/ Comments

`login.example.com:443`	`WEBHOST1.example.com:7777`
	`WEBHOST2.example.com:7777`
		HTTPS	Yes	Identity Management requires that the following be added to the HTTP header:
				`Header Name: IS_SSL`
				`Header Value: ssl`
				`Header Name: WL-Proxy-SSL`
				`Header Value: true`

Virtual Host	Server Pool	Protocol	SSL Termination?	Other Required Configuration/ Comments

`prov.example.com:443`	`WEBHOST1.example.com:7777`
	`WEBHOST2.example.com:7777`
		HTTPS		Identity Management requires that the following be added to the HTTP header:
				`Header Name: IS_SSL`
				`Header Value: ssl`
				`Header Name: WL-Proxy-SSL`
				`Header Value: true`

`iadadmin.example.com:80`	`WEBHOST1.example.com:7777` `WEBHOST2.example.com:7777`	HTTP		
`igdadmin.example.com:80`	`WEBHOST1.example.com:7777` `WEBHOST2.example.com:7777`	HTTP		
`igdinternal.example.com:7777`	`WEBHOST1.example.com:7777` `WEBHOST2.example.com:7777`	HTTP		
`idstore.example.com:1389`	`LDAPHOST1.example.com:1389` `LDAPHOST2.example.com:1389`	TCP		
`idstore.example.com:1636`	`LDAPHOST1.example.com:1636` `LDAPHOST2.example.com:1636`	TCP		
`oam.example.com:5575`	`OAMHOST1.example.com:5575` `OAMHOST2.example.com:5575`	TCP	No	Only required for active-active multi datacenter deployments.

Note:

- Port 80 is the `HTTP_PORT` from the Worksheet.
- Port 443 is the `HTTPS_PORT` from the Worksheet.
- Port 7777 is the `OHS_PORT` from the Worksheet.
- Port 1389 is the `LDAP_PORT` from the Worksheet. The example given is for OUD.
- Port 1636 is the `LDAP_SSL_PORT` from the worksheet. The example given is for OUD.
- Port 5575 is the `OAM_PROXY_PORT` from the worksheet.

Summary of the Virtual Servers Required for an Oracle Identity and Access Management Deployment

For an Oracle Identity and Access Management deployment, configure your load balancer as described.

Table 7-2 Load Balancer Configuration Details

Load Balancer Virtual Server	Server Pool	Server Pool (External OHS)	Protocol	SSL Termination	External	Other Required Configuration/Comments

`login.example.com:443`	`WEBHOST1vhn1.example.com:7777`
	`WEBHOST2vhn1.example.com:7777`
		`WEBHOST1.example.com:7777`
		`WEBHOST2.example.com:7777`
			HTTPS	Yes	Yes	Identity Management requires that the following be added to the HTTP header:
						`Header Name: IS_SSL`[1]
						`Header Value: ssl`
						`Header Name: WL-Proxy-SSL`
						`Header Value: true`

`prov.example.com:443`	`WEBHOST1vhn1.example.com:7777`
	`WEBHOST2vhn1.example.com:7777`
		`OHSHOST1.example.com:7777`
		`OHSHOST2.example.com:7777`
			HTTPS	Yes	Yes	Identity Management requires that the following be added to the HTTP header:
						`Header Name: IS_SSL`
						`Header Value: SSL`
						`Header Name: WL-Proxy-SSL`
						`Header Value: true`

`iadadmin.example.com:80`	`WEBHOST1vhn1.example.com:7777` `WEBHOST2vhn1.example.com:7777`	`OHSHOST1.example.com:7777` `OHSHOST2.example.com:7777`	HTTP	No	No	NA
`igdadmin.example.com:80`	`WEBHOST1vhn1.example.com:7777` `WEBHOST2vhn1.example.com:7777`	`OHSHOST1.example.com:7777` `OHSHOST2.example.com:7777`	HTTP	No	No	NA

Table 7-2 (Cont.) Load Balancer Configuration Details

Load Balancer Virtual Server	Server Pool	Server Pool (External OHS)	Protocol	SSL Termination	External	Other Required Configuration/Comments
`oam.example.com:5575`	`WEBHOST1vhn1.example.com:7777` `WEBHOST2vhn1.example.com:7777`	`OAMHOST1.example.com:5575` `OAMHOST2.example.com:5575`	TCP	No	No	Only required for active-active multi datacenter deployments.

[1] For information about configuring IS_SSL, see About User Defined WebGate Parameters in *Administrator's Guide for Oracle Access Management*.

If you are using an external OHS then the servers will point to the external OHS hosts.

For information about configuring IS_SSL, see About User Defined WebGate Parameters in *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management*.

Note:

Port 80 is the *HTTP_PORT* from the Worksheet

Port 443 is the *HTTPS_PORT* from the Worksheet

Port 7777 is the *OHS_PORT* from the Worksheet

Port 1389 is the *LDAP_PORT* from the Worksheet

Port 1636 is the *LDAP_SSL_PORT* from the worksheet

Configuring Global Load Balancers

As indicated in the previous sections, the Global Load Balancer (GLBR) is responsible for performing smart routing of requests between multiple Local Load Balancers.

This smart routing is usually done based on the originating request. In an Oracle Fusion Middleware Identity and Access Management multi-data center active-active deployment, it is recommended that you restrain callbacks and invocations that come from servers in a specific site to the same site again. As the GLBR is typically located in one of the two sites, physically, this also makes the invocations to such a site more efficient. You must configure a global load balancer for multi-site deployments whether the second site is active or being used for disaster recovery.

Configuring the Firewalls and Ports for an Enterprise Deployment

As an administrator, it is important that you become familiar with the port numbers that are used by various Oracle Fusion Middleware products and services. This ensures that the same port number is not used by two services on the same host, and that the proper ports are open on the firewalls in the enterprise topology.

The following tables lists the ports that you must open on the firewalls in the topology:

Firewall notation:

- FW0 refers to the outermost firewall.
- FW1 refers to the firewall between the web tier and the application tier.
- FW2 refers to the firewall between the application tier and the data tier.

Table 7-3 Firewall Ports Common to All Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Browser request	FW0	80	HTTP / Load Balancer	Inbound	Timeout depends on the size and type of HTML content.
Browser request	FW0	443	HTTPS / Load Balancer	Inbound	Timeout depends on the size and type of HTML content.
Browser request	FW1	80	HTTP / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Browser request	FW1	443	HTTPS / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	80	HTTP / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	443	HTTPS / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Load balancer to Oracle HTTP Server	n/a	7777	HTTP	n/a	n/a
OHS registration with Administration Server	FW1	7001	HTTP / t3	Inbound	Set the timeout to a short period (5-10 seconds).
OHS management by Administration Server	FW1	OHS Admin Port (7779)	TCP / HTTP	Outbound	Set the timeout to a short period (5-10 seconds).
Session replication within a WebLogic Server cluster	n/a	n/a	n/a	n/a	By default, this communication uses the same port as the server's listen address.

Table 7-3 (Cont.) Firewall Ports Common to All Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Administration Console access	FW1	7001	HTTP / Administration Server and Enterprise Manager t3	Both	You should tune this timeout based on the type of access to the admin console (whether you plan to use the Oracle WebLogic Server Administration Console from the application tier clients or clients external to the application tier).
Database access	FW2	1521	SQL*Net	Both	Timeout depends on database content and on the type of process model used for SOA.
Coherence for deployment	n/a	9991	n/a	n/a	n/a
Oracle Unified Directory access	FW2	389 636 (SSL)	LDAP or LDAP/ssl	Inbound	You should tune the directory server's parameters based on load balancer, and not the other way around.
Oracle Notification Server (ONS)	FW2	6200	ONS	Both	Required for Gridlink. An ONS server runs on each database server.

Table 7-4 Firewall Ports Specific to the Oracle Identity and Access Management Enterprise Deployment

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Webtier Access to Oracle Weblogic Administration Server (IAMAccessDomain)	FW1	7010	HTTP / Oracle HTTP Server and Administration Server	Inbound	N/A

Table 7-4 (Cont.) Firewall Ports Specific to the Oracle Identity and Access Management Enterprise Deployment

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Webtier Access to Oracle Weblogic Administration Server (IAMGovernanceDomain)	FW1	7101	HTTP / Oracle HTTP Server and Administration Server	Inbound	N/A
WSM-PM access	FW1	7010 Range: 7010 to 7999	HTTP / WLS_WSM-PMn	Inbound	Set the timeout to 60 seconds.
Enterprise Manager Agent - web tier to Enterprise Manager	FW1	5160	HTTP / Enterprise Manager Agent and Enterprise Manager	Both	N/A
Oracle HTTP Server to WLS_OAM	FW1	14100	HTTP / Oracle HTTP Server to WebLogic Server	Inbound	Timeout depends on the `mod_weblogic` parameters used
Oracle HTTP Server WLS_OIM	FW1	14000	HTTP / Oracle HTTP Server to WebLogic Server	Inbound	Timeout depends on the `mod_weblogic` parameters used
Oracle HTTP Server WLS_SOA	FW1	8001	HTTP / Oracle HTTP Server to WebLogic Server	Both	Timeout depends on the `mod_weblogic` parameters used
Oracle HTTP Server WLS_AMA	FW1	14150	HTTP / Oracle HTTP Server to WebLogic Server	Both	Timeout depends on the `mod_weblogic` parameters used
Oracle HTTP Server WLS_BI	FW1	9704	HTTP / Oracle HTTP Server to WebLogic Server	Both	Timeout depends on the `mod_weblogic` parameters used
Access Manager Server	FW1	5575	OAP	Both	N/A
Access Manager Coherence port	FW1	9095	TCMP	Both	N/A
Oracle Coherence Port	FW1	8000–8088	TCMP	Both	N/A

Table 7-4 (Cont.) Firewall Ports Specific to the Oracle Identity and Access Management Enterprise Deployment

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Application Tier to Database Listener	FW2	1521	SQL*Net	Both	Timeout depends on all database content and on the type of process model used for Oracle Identity and Access Management
Oracle Notification Server (ONS)	FW2	6200	ONS	Both	Required for Gridlink. An ONS server runs on each database server
OUD Port	FW2	1389	LDAP	Inbound	Ideally, these connections should be configured not to time out
OUD SSL Port	FW2	14636	LDAPS	Inbound	Ideally, these connections should be configured not to time out
Load Balancer LDAP Port	FW2	386	LDAP	Inbound	Ideally, these connections should be configured not to time out
Load Balancer LDAP SSL Port	FW2	636	LDAPS	Inbound	Ideally, these connections should be configured not to time out
Node Manager	N/A	5556	TCP/IP	N/A	N/A
Oracle Unified Directory Replication	N/A	8989	TCP/IP	N/A	N/A

8

Preparing the File System for an Enterprise Deployment

Preparing the file system for an enterprise deployment involves understanding the requirements for local and shared storage, as well as the terminology that is used to reference important directories and file locations during the installation and configuration of the enterprise topology.

This chapter describes how to prepare the file system for an Oracle Fusion Middleware enterprise deployment.

- Overview of Preparing the File System for an Enterprise Deployment
  It is important to set up your storage in a way that makes the enterprise deployment easy to understand, configure, and manage.
- Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment
  Oracle recommends that you implement certain guidelines regarding shared storage when you install and configure an enterprise deployment.
- About the Recommended Directory Structure for an Enterprise Deployment
  The diagrams in this section show the recommended directory structure for a typical Oracle Fusion Middleware enterprise deployment.
- File System and Directory Variables Used in This Guide
  Understanding the file system directories and the directory variables used to reference these directories is essential for installing and configuring the enterprise deployment topology.
- About Creating and Mounting the Directories for an Enterprise Deployment
  Oracle recommends that you implement certain best practices when you create or mount the top-level directories in an enterprise deployment.
- Summary of the Shared Storage Volumes in an Enterprise Deployment
  It is important to understand the shared volumes and their purpose in a typical Oracle Fusion Middleware enterprise deployment.

Overview of Preparing the File System for an Enterprise Deployment

It is important to set up your storage in a way that makes the enterprise deployment easy to understand, configure, and manage.

This chapter provides an overview of the process of preparing the file system for an enterprise deployment. Oracle recommends setting up your storage according to information in this chapter. The terminology defined in this chapter is used in the diagrams and procedures throughout the guide.

Use this chapter as a reference to understand the directory variables that are used in the installation and configuration procedures.

Other directory layouts are possible and supported, but the model adopted in this guide was designed for maximum availability, providing both the best isolation of components and symmetry in the configuration and facilitating backup and disaster recovery. The rest of the document uses this directory structure and directory terminology.

Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment

Oracle recommends that you implement certain guidelines regarding shared storage when you install and configure an enterprise deployment.

Before you implement the detailed recommendations in this chapter, be sure to review the recommendations and general information about using shared storage in the *High Availability Guide*.

The recommendations in this chapter are based on the concepts and guidelines described in the *High Availability Guide*.

Table 8-1 lists the key sections that you should review and how those concepts apply to an enterprise deployment.

Table 8-1 Shared Storage Resources in the High Availability Guide

Section in *High Availability Guide*	Importance to an Enterprise Deployment
Shared Storage Prerequisites	Describes guidelines for disk format and the requirements for hardware devices that are optimized for shared storage.
Using Shared Storage for Binary (Oracle Home) Directories	Describes your options for storing the Oracle home on a shared storage device that is available to multiple hosts. For an enterprise deployment, Oracle recommends that you use redundant Oracle homes on separate storage volumes. If a separate volume is not available, a separate partition on the shared disk should be used to provide redundant Oracle homes to application tier hosts.
Using Shared Storage for Domain Configuration Files	Describes the concept of creating separate domain homes for the Administration Server and the Managed Servers in the domain. For an enterprise deployment, the Administration Server domain home location is referenced by the *ASERVER_HOME* variable.
Shared Storage Requirements for JMS Stores and JTA Logs	Provides instructions for setting the location of the transaction logs and JMS stores for an enterprise deployment.
Introduction to Zero Downtime Patching	Describes the Zero Downtime feature and the procedure to configure and monitor workflows.

> **Note:**
>
> Zero Downtime Patching (ZDT Patching) provides an automated mechanism to orchestrate the rollout of patches while avoiding downtime or loss of sessions. ZDT reduces risks and downtime of mission-critical applications that require availability and predictability while applying patches.
>
> By using the workflows that you define, you can patch or update any number of nodes in a domain with little or no manual intervention. Changes are rolled out to one node at a time. This preemptively allows for session data to be migrated to compatible servers in the cluster and allows service migration of singleton services, such as JTA and JMS.
>
> When you patch the Oracle home, the current Oracle home must be installed locally on each node that is included in the workflow. Although it is not required, Oracle also recommends that the Oracle home be in the same location on each node.

About the Recommended Directory Structure for an Enterprise Deployment

The diagrams in this section show the recommended directory structure for a typical Oracle Fusion Middleware enterprise deployment.

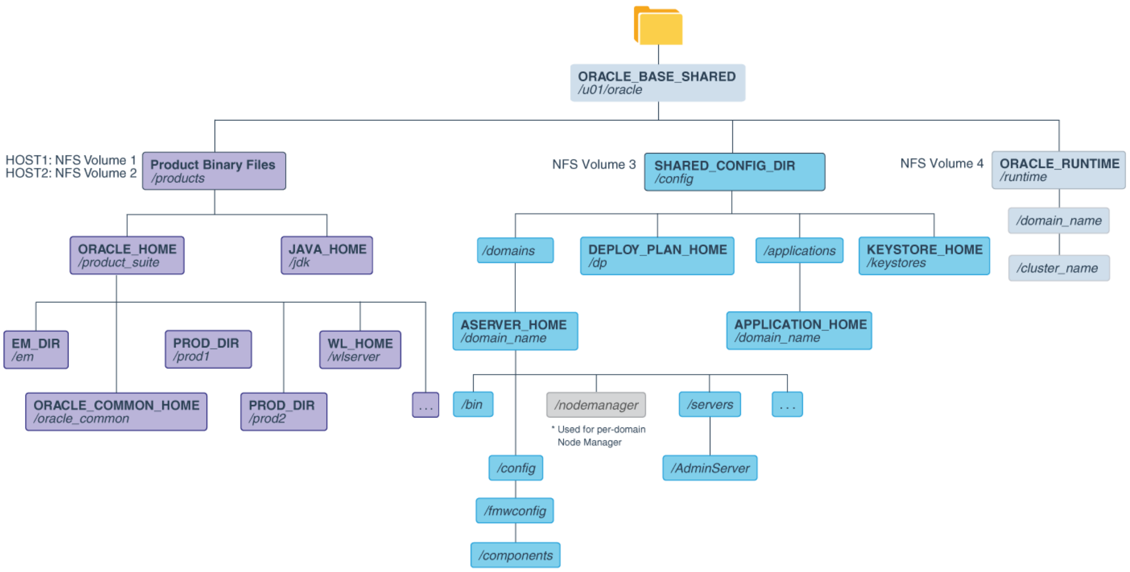
The directories shown in the diagrams contain binary files that are installed on the disk by the Oracle Fusion Middleware installers, domain-specific files generated through the domain configuration process, as well as domain configuration files that are propagated to the various host computers through the Oracle WebLogic Server `pack` and `unpack` commands.

The diagrams are used to indicate:

- Figure 8-1 shows the resulting directory structure on the shared storage device after you have installed and configured a typical Oracle Fusion Middleware enterprise deployment. The shared storage directories are accessible by the application tier host computers.
- Figure 8-2 shows the resulting directory structure on the local storage device for a typical application tier host after you have installed and configured an Oracle Fusion Middleware enterprise deployment. The Managed Servers in particular are stored on the local storage device for the application tier host computers.
- Figure 8-4 shows the resulting directory structure on the local storage device for a typical web tier host after you have installed and configured an Oracle Fusion Middleware enterprise deployment. Note that the software binaries (in the Oracle home) are installed on the local storage device for each web tier host.

Where applicable, the diagrams also include the standard variables used to reference the directory locations in the installation and configuration procedures in this guide.

Figure 8-1 Recommended Shared Storage Directory Structure for an Enterprise Deployment

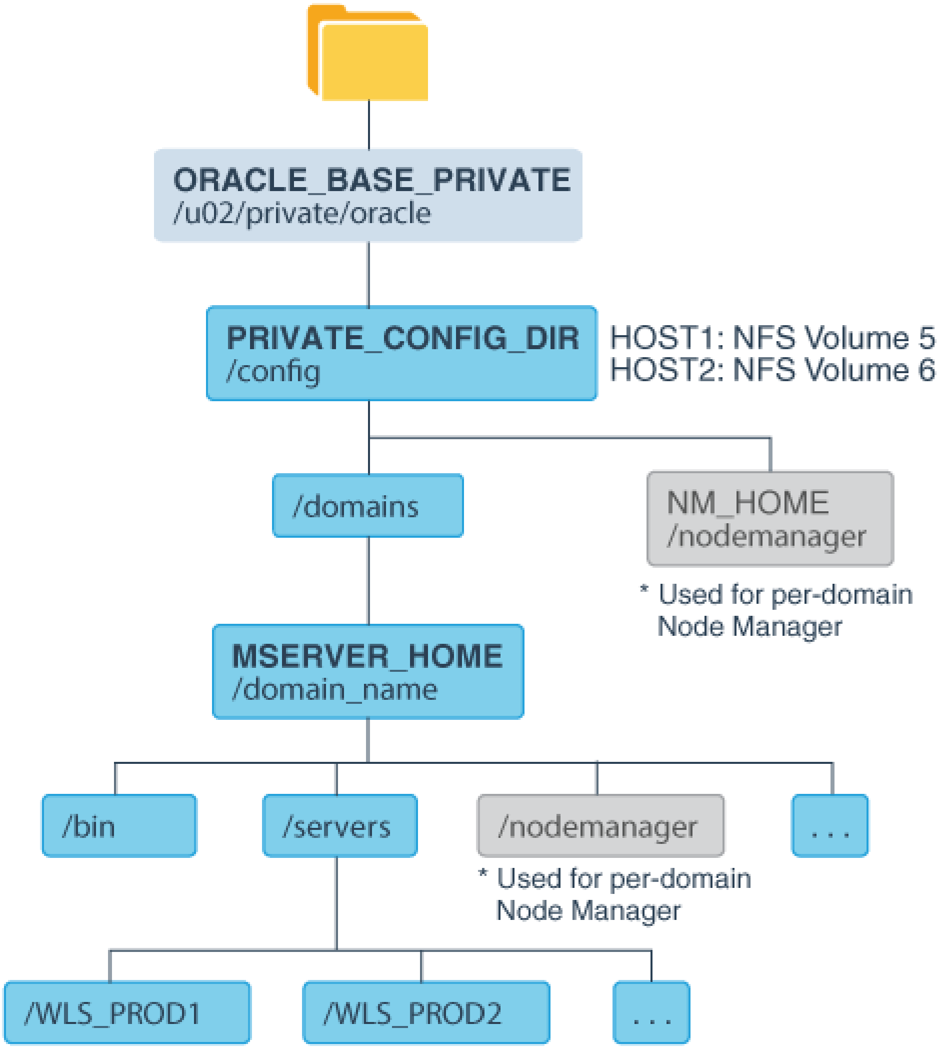


Note:

- In a deployment which uses more than one domain, it is recommended that different copies of the binaries are used for each domain. For Oracle Identity Management, it is recommended that the following ORACLE_HOMEs be created:
  - `DIR_ORACLE_HOME` for Directory components (`/dir`)
  - `IAD_ORACLE_HOME` for Access components (`/access`)
  - `IGD_ORACLE_HOME` for Identity Governance components (`/identity`)
- Oracle Identity Management uses multiple Domains. The diagram above depicts ASERVER_HOME as a generic location. This guide will often prefix this value with a domain abbreviation for example:
  - `IAD_ASERVER_HOME` for Oracle Access Manager
  - `IGD_ASERVER_HOME` for Oracle identity Governance

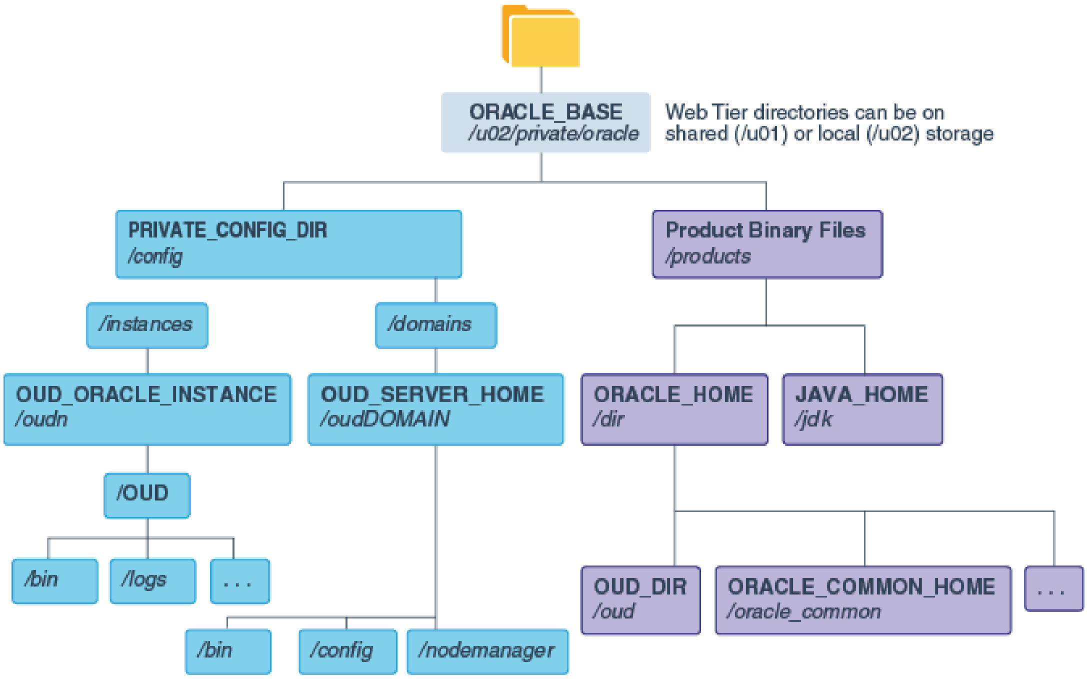
*See About the Node Manager Configuration in a Typical Enterprise Deployment.

Figure 8-2 Recommended Local Storage Directory Structure for an Application Tier Host Computer in an Enterprise Deployment



See About the Node Manager Configuration in a Typical Enterprise Deployment.

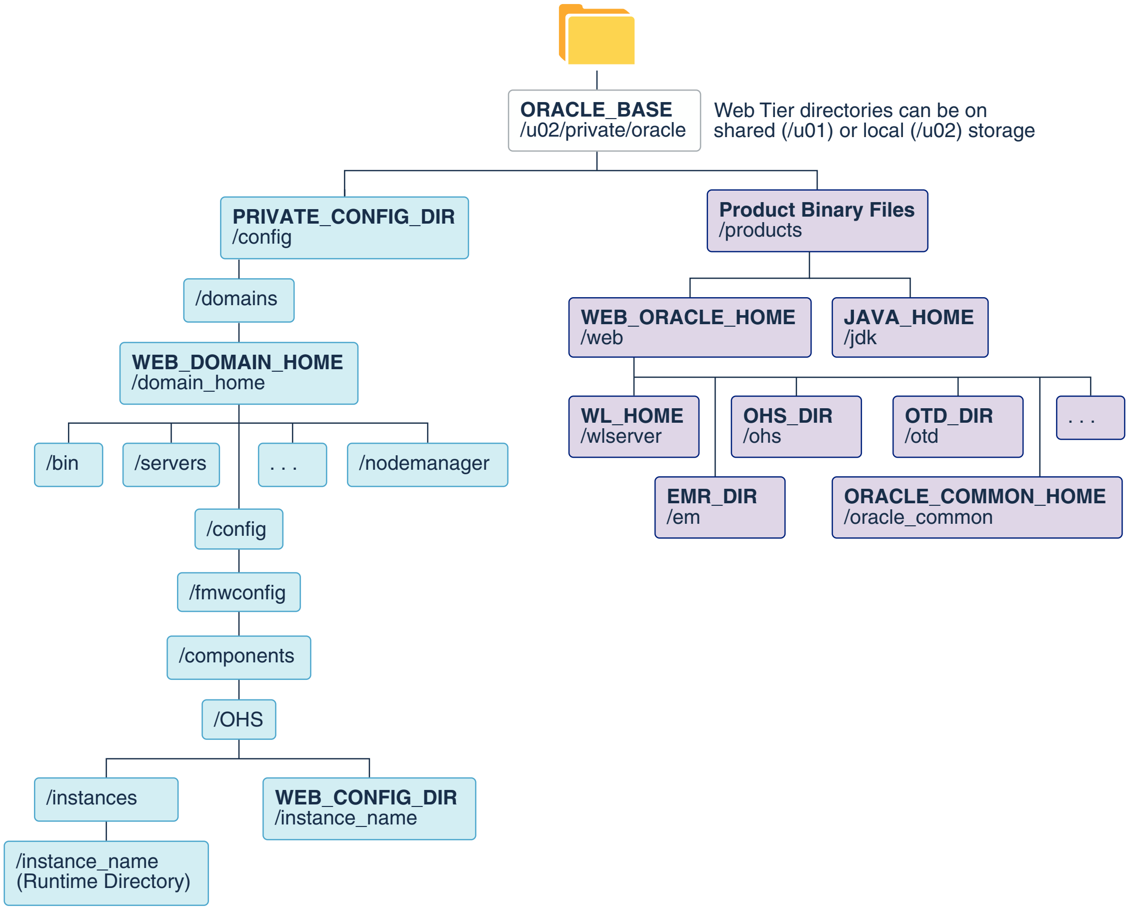
Figure 8-3 Recommended Local Storage Directory Structure for a Directory Tier Host Computer in an Enterprise Deployment



Note:

`oudDomain` exists only if OUDSM is used.

Figure 8-4 Recommended Local Storage Directory Structure for a Web Tier Host Computer in an Enterprise Deployment



File System and Directory Variables Used in This Guide

Understanding the file system directories and the directory variables used to reference these directories is essential for installing and configuring the enterprise deployment topology.

Table 8-2 lists the file system directories and the directory variables that are used to reference the directories on the application tier. Table 8-3 lists the file system directories and variables that are used to reference the directories on the web tier.

For additional information about mounting these directories when you use shared storage, see About Creating and Mounting the Directories for an Enterprise Deployment.

Throughout this guide, the instructions for installing and configuring the topology refer to the directory locations that use the variables shown here.

You can also define operating system variables for each of the directories listed in this section. If you define system variables for the particular UNIX shell that you are using, you can then use the variables as they are used in this document, without having to map the variables to the actual values for your environment.

Note:

As you configure your storage devices to accommodate the recommended directory structure, note the actual directory paths in the Enterprise Deployment workbook. You will use these addresses later when you enable the IP addresses on each host computer.

See Using the Enterprise Deployment Workbook.

Table 8-2 Sample Values for Key Directory Variables on the Application Tier

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
ORACLE_BASE	The base directory, under which Oracle products are installed.	N/A	`/u01/oracle`

ORACLE_HOME	The read-only location for the product binaries. For the application tier host computers, it is stored on shared disk.
	The Oracle home is created when you install the Oracle Fusion Middleware Infrastructure software.
	You can then install additional Oracle Fusion Middleware products into the same Oracle home.
		ORACLE_BASE/products/product	`/u01/oracle/products/access`
			`/u01/oracle/products/identity`
			`/u01/oracle/products/dir`

ORACLE_COMMON_HOME	The directory within the Oracle Fusion Middleware Oracle home where common utilities, libraries, and other common Oracle Fusion Middleware products are stored.	*ORACLE_HOME*/oracle_common	`/oracle_common/u01/oracle/products/access`
			`/oracle_common/u01/oracle/products/dir`
			`/oracle_common/u01/oracle/products/identity`

WL_HOME	The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored.	*ORACLE_HOME*/wlserver	`/u01/oracle/products/product/wlserver`
PROD_DIR	Individual product directories for each Oracle Fusion Middleware product that you install.	*ORACLE_HOME*/*prod_dir*	/u01/oracle/products/product/*prod_dir* The product can be `soa`, `wcc`, `idm`, `bi`, or another value, depending on your enterprise deployment.
EM_DIR	The product directory used to store the Oracle Enterprise Manager Fusion Middleware Control software binaries.	*ORACLE_HOME*/em	`/u01/oracle/products/product/em`
JAVA_HOME	The location where you install the supported Java Development Kit (JDK).	*ORACLE_BASE*/products/jdk	`/u01/oracle/products/jdk`

Table 8-2 (Cont.) Sample Values for Key Directory Variables on the Application Tier

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
SHARED_CONFIG_DIR	The shared parent directory for shared environment configuration files, including domain configuration, keystores, runtime artifacts, and application deployments	*ORACLE_BASE*/config	/u01/oracle/config
PRIVATE_CONFIG_DIR	The local or nfs-mounted private configuration directory unique to a given host containing the machine-specific domain directory (MSERVER_HOME). Directory variable: PRIVATE_CONFIG_DIR	/u02/private/oracle/config	/u02/private/oracle/config
ASERVER_HOME	The Administration Server domain home, which is installed on a shared disk.	*SHARED_CONFIG_DIR*/domains/*domain_name*	/u01/oracle/config/domains/*domain_name* In this example, replace *domain_name* with the name of the WebLogic Server domain.
MSERVER_HOME	The Managed Server domain home, which is created by using the unpack command on the local disk of each application tier host.	*PRIVATE_CONFIG_DIR*/domains/*domain_name*	/u02/private/oracle/config/domains/*domain_name* In this example, replace *domain_name* with the name of the WebLogic Server domain.
APPLICATION_HOME	The Application home directory, which is installed on shared disk, so the directory is accessible by all the application tier host computers.	*SHARED_CONFIG_DIR*/applications/*domain_name*	/u01/oracle/config/applications/*domain_name* In this example, replace *domain_name* with the name of the WebLogic Server domain.

Table 8-2 (Cont.) Sample Values for Key Directory Variables on the Application Tier

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
ORACLE_RUNTIME	This directory contains the Oracle runtime artifacts, such as the JMS logs and TLogs. Typically, you mount this directory as a separate shared file system, which is accessible by all hosts in the domain. When you run the Configuration Wizard or perform post-configuration tasks, and you identify the location of JMS stores or tlogs persistent stores, then you can use this directory, qualified with the name of the domain, the name of the cluster, and the purpose of the directory. For example: `ORACLE_RUNTIME/cluster_name/jms`	`ORACLE_BASE/runtime`	`/u01/oracle/runtime/`
NM_HOME	The directory used by the Per Machine Node Manager start script and configuration files. **Note:** This directory is necessary only if you are using a Per Machine Node Manager configuration. See About the Node Manager Configuration in a Typical Enterprise Deployment.	`PRIVATE_CONFIG_DIR/node_manager`	`/u02/private/oracle/config/node_manager`
DEPLOY_PLAN_HOME	The deployment plan directory, which is used as the default location for application deployment plans. **Note:** This directory is required only when you are deploying custom applications to the application tier.	`SHARED_CONFIG_DIR/dp`	`/u01/oracle/config/dp`
KEYSTORE_HOME	The shared location for custom certificates and keystores.	`SHARED_CONFIG_DIR/keystores`	`/u01/oracle/config/keystores`

Table 8-3 Sample Values for Key Directory Variables on the Web Tier

Directory Variable	Description	Sample Value on the Web Tier
WEB_ORACLE_HOME	The read-only location for the Oracle HTTP Server product binaries. For the web tier host computers, this directory is stored on the local disk. The Oracle home is created when you install the Oracle HTTP Server software .	`/u02/private/oracle/products/web`

Table 8-3 (Cont.) Sample Values for Key Directory Variables on the Web Tier

Directory Variable	Description	Sample Value on the Web Tier
ORACLE_COMMON_HOME	The directory within the Oracle HTTP Server Oracle home where common utilities, libraries, and other common Oracle Fusion Middleware products are stored.	`/u02/private/oracle/products/web/oracle_common`
WL_HOME	The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored.	`/u02/private/oracle/products/web/wlserver`
PROD_DIR	Individual product directories for each Oracle Fusion Middleware product that you install.	`/u02/private/oracle/products/web/ohs`
JAVA_HOME	The location where you install the supported Java Development Kit (JDK).	`/u02/private/oracle/products/jdk`
WEB_DOMAIN_HOME	The Domain home for the standalone Oracle HTTP Server domain, which is created when you install Oracle HTTP Server on the local disk of each web tier host.	`/u02/private/oracle/config/domains/`*`domain_name`* In this example, replace *`domain_name`* with the name of the WebLogic Server domain.
WEB_CONFIG_DIR	This is the location where you edit the Oracle HTTP Server configuration files (for example, `httpd.conf` and `moduleconf/*.conf`) on each web host. Note that this directory is also referred to as the OHS Staging Directory. Changes made here are later propagated to the OHS Runtime Directory. See Staging and Run-time Configuration Directories in the *Administering Oracle HTTP Server*.	

About Creating and Mounting the Directories for an Enterprise Deployment

Oracle recommends that you implement certain best practices when you create or mount the top-level directories in an enterprise deployment.

- For the application tier, install the Oracle home, which contains the software binaries, on a second shared storage volume or second partition that is mounted to OAMHOST2. Be sure the directory path to the binaries on OAMHOST2 is identical to the directory path on OAMHOST1.

  For example:

  ```
  /u01/oracle/products/product_suite/
  ```

  See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

- This enterprise deployment guide assumes that the Oracle Web tier software is installed on a local disk.

  The Web tier installation is typically performed on local storage to the WEBHOST nodes. When you use shared storage, you can install the Oracle Web tier binaries (and create the Oracle HTTP Server instances) on a shared disk. However, if you do so, then the shared disk *must* be separate from the shared disk used for the application tier, and you must consider the appropriate security restrictions for access to the storage device across tiers.

As with the application tier servers (OAMHOST1 and OAMHOST2), use the same directory path on both computers.

For example:

```
/u02/private/oracle/products/web
```

Summary of the Shared Storage Volumes in an Enterprise Deployment

It is important to understand the shared volumes and their purpose in a typical Oracle Fusion Middleware enterprise deployment.

The following table summarizes the shared volumes and their purpose in a typical Oracle Fusion Middleware enterprise deployment.

See, Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

Table 8-4 Shared Storage Volumes in an Enterprise Deployment

Shared Volume Name	Mounted to Host	Mount Directories	Description and Purpose
Binaries1	OAMHOST1 OIMHOST1 LDAPHOST1	`/u01/oracle/products/`	Shared storage for the product binaries to be used. This is where the Oracle home directory and product directories are installed.
Binaries2	OAMHOST2 OIMHOST2 LDAPHOST2	`/u01/oracle/products/`	Shared storage for the product binaries to be used. This is where the Oracle home directory and product directories are installed.
sharedConfig	OAMHOST1 OAMHOST2 OIMHOST1 OIMHOST2	`/u01/oracle/config/`	Administration Server domain configuration, mounted to all hosts; used initially by HOST1, but can be failed over to any host.
runTime	OIMHOST1 OIMHOST2	`/u01/oracle/runtime/`	The runtime artifacts directory, mounted to all hosts, contains runtime artifacts such as JMS logs, blogs, and any cluster-dependent shared files needed.
webBinaries1	WEBHOST1	`/u02/private/oracle/products`	Local storage for the Oracle HTTP Server software binaries (Oracle home).
webBinaries2	WEBHOST2	`/u02/private/oracle/products`	Local storage for the Oracle HTTP Server software binaries (Oracle home).

Table 8-4 (Cont.) Shared Storage Volumes in an Enterprise Deployment

Shared Volume Name	Mounted to Host	Mount Directories	Description and Purpose
webConfig1	WEBHOST1	`/u02/private/oracle/config`	Local storage for the domain configuration files used by WEBHOST1, if the private Managed Server domain directory resides on shared storage.
webConfig2	WEBHOST2	`/u02/private/oracle/config`	Local storage for the domain configuration files used by WEBHOST2, if the private Managed Server domain directory resides on shared storage.

Note:

Directory Binaries can either be created locally or shared. If shared, create a separate volume for the Directory binaries.

If Shared storage is being used for local storage, to make the backups easier or to have the files stored on more fault tolerant hardware, then you must create an NFS volume for each of the hosts in the topology. It should then be mounted to the host in the location *ORACLE_BASE*.

9 Preparing the Operating System for an Oracle Identity and Access Management Deployment

Preparing the operating system consists of performing all of the previous preparatory steps. Once completed, the environment will have the same structure as a traditional server deployment.

This chapter contains the following sections:

- Summary of Storage Requirements

Summary of Storage Requirements

This section summarizes storage requirements for an Oracle Identity and Access Management deployment.

- Summary of the Storage Appliance Directories and Corresponding Mount Points for Virtual Servers

Summary of the Storage Appliance Directories and Corresponding Mount Points for Virtual Servers

For the Oracle Identity Management enterprise topology, you install all software products on the shared file system.

To organize the enterprise deployment software on the appliance, you create a new project, called `IAM`. The shares (`/products` and `/config`) are created within this project on the appliance, so you can later mount the shares to each compute node.

To separate the product binaries from the files specific to each compute node, you create a separate share for each compute node. Sub-directories are for the host names are created under `config` and `products` directories. Each private directory is identified by the logical host name; for example, `IAMHOST1` and `IAMHOST2`.

Figure 9-1 shows the recommended physical directory structure on the .

Table 9-1 shows how the shares on the appliance map to the mount points you will create on the vServers that host the enterprise deployment software.

Figure 9-1 Physical Structure of the Shares on the Shared File System for Virtual Deployments

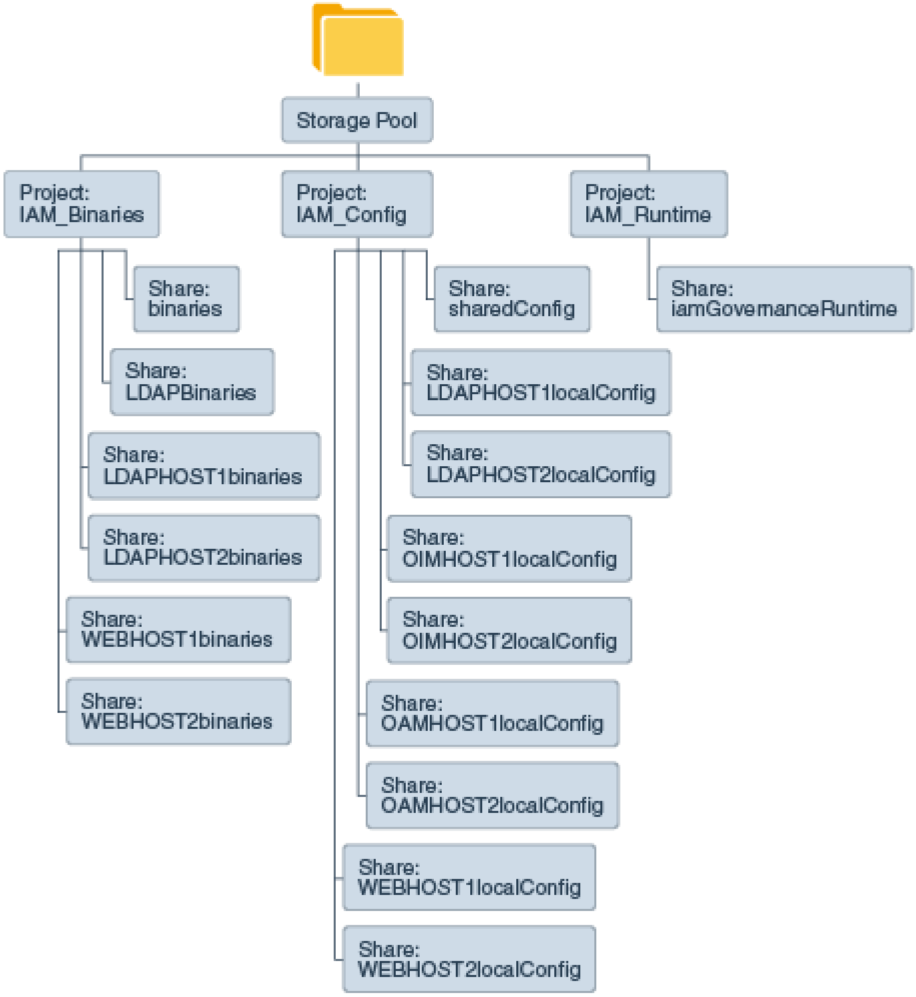

Storage Pool
Project: IAM_Binaries
Project: IAM_Config
Project: IAM_Runtime
Share: binaries
Share: LDAPBinaries
Share: LDAPHOST1binaries
Share: LDAPHOST2binaries
Share: WEBHOST1binaries
Share: WEBHOST2binaries
Share: sharedConfig
Share: LDAPHOST1localConfig
Share: LDAPHOST2localConfig
Share: OIMHOST1localConfig
Share: OIMHOST2localConfig
Share: OAMHOST1localConfig
Share: OAMHOST2localConfig
Share: WEBHOST1localConfig
Share: WEBHOST2localConfig
Share: iamGovernanceRuntime

Figure 9-1 illustrates the physical structure of the shares on the shared appliance.

Table 9-1 Mapping the Shares on the Appliance to Mount Points on Each vServer

Project	Share	Mount Point	Host	Mounted On	Privileges to Assign to User, Group, and Other	Actual Size
IAM_Binaries	binaries	/export/IAM_Binaries/binaries	OAMHOST1 OAMHOST2 OIMHOST1 OIMHOST2	/u01/oracle/products	R and W (Read and Write)	35 GB
IAM_Binaries	LDAPBinaries	/export/IAM_Binaries/LDAPBinaries	LDAPHOST1 LDAPHOST2	/u01/oracle/products	R and W (Read and Write)	10 GB

Table 9-1 (Cont.) Mapping the Shares on the Appliance to Mount Points on Each vServer

Project	Share	Mount Point	Host	Mounted On	Privileges to Assign to User, Group, and Other	Actual Size
IAM_Binaries	WEBHOST1binaries	/export/IAM_Binaries/webhost1binaries	WEBHOST1	/u01/oracle/products	R and W (Read and Write)	10 GB
IAM_Binaries	WEBHOST2binaries	/export/IAM_Binaries/webhost2binaries	WEBHOST2	/u01/oracle/products	R and W (Read and Write)	10 GB
IAM_Config	sharedConfig	/export/IAM_Config/sharedConfig	OAMHOST1 OAMHOST2 OIMHOST1 OIMHOST2	/u01/oracle/config	R and W (Read and Write)	100 GB
IAM_Config	OAMHOST1localConfig	/export/IAM_Config/oamhost1localConfig	OAMHOST1	/u02/private/oracle/config	R and W (Read and Write)	10 GB
IAM_Config	OAMHOST2localConfig	/export/IAM_Config/oamhost2localConfig	OAMHOST2	/u02/private/oracle/config	R and W (Read and Write)	10 GB
IAM_Config	OIMHOST1localConfig	/export/IAM_Config/oimhost1localConfig	OIMHOST1	/u02/private/oracle/config	R and W (Read and Write)	80 GB
IAM_Config	OIMHOST2localConfig	/export/IAM_Config/oimhost2localConfig	OIMHOST2	/u02/private/oracle/config	R and W (Read and Write)	80 GB
IAM_Config	WEBHOST1localConfig	/export/IAM_Config/webhost1localConfig	WEBHOST1	/u02/private/oracle/config	R and W (Read and Write)	5 GB
IAM_Config	WEBHOST2localConfig	/export/IAM_Config/webhost2localConfig	WEBHOST2	/u02/private/oracle/config	R and W (Read and Write)	5 GB
IAM_Config	LDAPHOST1localConfig	/export/IAM_Config/ldaphost1localConfig	LDAPHOST1	/u02/private/oracle/config	R and W (Read and Write)	5 GB
IAM_Config	LDAPHOST2localConfig	/export/IAM_Config/ldaphost2localConfig	LDAPHOST2	/u02/private/oracle/config	R and W (Read and Write)	5 GB

Table 9-1 (Cont.) Mapping the Shares on the Appliance to Mount Points on Each vServer

Project	Share	Mount Point	Host	Mounted On	Privileges to Assign to User, Group, and Other	Actual Size
IAM_Runtime	`iamGovernanceRuntime`	`/export/IAM_Runtime/iamGovernanceRuntime`	OIMHOST1 OIMHOST2	`/u01/oracle/runtime`	R and W (Read and Write)	5 GB

Note:

The `binary` directories can be changed to **read only** after the configuration is complete if desired. The LDAPHOST binaries have been split into two shares, one for each node. These can be combined, if required.

Table 9-2 Summary of Storage Projects for Virtual Servers

Project	Size
IAM_Binaries	100 GB
IAM_Config	300 GB
IAM_Runtime	5 GB

10

Preparing the Host Computers for an Enterprise Deployment

It is important to perform a set of tasks on each computer or server before you configure the enterprise deployment topology. This involves verifying the minimum hardware and operating system requirements for each host, configuring operating system users and groups, enabling Unicode support, mounting the required shared storage systems to the host and enabling the required virtual IP addresses on each host.

This chapter describes the tasks that you must perform from each computer or server that is hosting the enterprise deployment.

Verifying the Minimum Hardware Requirements for Each Host

After you procure the required hardware for the enterprise deployment, it is important to ensure that each host computer meets the minimum system requirements.

After you have procured the required hardware for the enterprise deployment, log in to each host computer and verify the system requirements listed in Hardware and Software Requirements for the Enterprise Deployment Topology.

Ensure that you have sufficient local disk storage and shared storage configured as described in Preparing the File System for an Enterprise Deployment.

Allow sufficient swap and temporary space; specifically:

- **Swap Space**–The system must have at least 500 MB.

- **Temporary Space**–There must be a minimum of 500 MB of free space in the `/tmp` directory.

Verifying Linux Operating System Requirements

You can review the typical Linux operating system settings for an enterprise deployment in this section.

To ensure the host computers meet the minimum operating system requirements, ensure that you have installed a certified operating system and that you have applied all the necessary patches for the operating system.

In addition, review the following sections for typical Linux operating system settings for an enterprise deployment.

- Setting Linux Kernel Parameters
- Setting the Open File Limit and Number of Processes Settings on UNIX Systems
- Verifying IP Addresses and Host Names in DNS or Hosts File

Setting Linux Kernel Parameters

The kernel-parameter and shell-limit values shown in Table 10-1 are recommended values only. Oracle recommends that you tune these values to optimize the performance of the system. See your operating system documentation for more information about tuning kernel parameters.

Kernel parameters must be set to a minimum of those in Table 10-1 on all nodes in the topology.

The values in the following table are the current Linux recommendations. For the latest recommendations for Linux and other operating systems, see *Oracle Fusion Middleware System Requirements and Specifications*.

If you deploy a database onto the host, you might need to modify additional kernel parameters. See Configuring Kernel Parameters in *Oracle Grid Infrastructure Installation Guide for Linux*.

Table 10-1 UNIX Kernel Parameters

Parameter	Value
kernel.sem	256 32000 100 142
kernel.shmmax	4294967295

To set these parameters:

1. Sign in as `root` and add or amend the entries in the `/etc/sysctl.conf` file.
2. Save the file.
3. Activate the changes by entering the following command:

   ```
   /sbin/sysctl -p
   ```

Setting the Open File Limit and Number of Processes Settings on UNIX Systems

On UNIX operating systems, the `Open File Limit` is an important system setting, which can affect the overall performance of the software running on the host computer.

For guidance on setting the `Open File Limit` for an Oracle Fusion Middleware enterprise deployment, see Host Computer Hardware Requirements.

> **Note:**
>
> The following examples are for Linux operating systems. Consult your operating system documentation to determine the commands to be used on your system.

For more information, see the following sections.

- Viewing the Number of Currently Open Files
- Setting the Operating System Open File and Processes Limits

Viewing the Number of Currently Open Files

You can see how many files are open with the following command:

```
/usr/sbin/lsof | wc -l
```

To check your open file limits, use the following commands.

C shell:

```
limit descriptors
```

Bash:

```
ulimit -n
```

Setting the Operating System Open File and Processes Limits

To change the Open File Limit values on Oracle Enterprise Linux 6 or greater:

1. Sign in as `root` user and edit the following file:

   ```
   /etc/security/limits.d/*-nproc.conf
   ```

   For example:

   ```
   /etc/security/limits.d/20-nproc.com
   ```

   > **Note:**
   >
   > The number can vary from host to host.

2. Add the following lines to the file. (The values shown here are for example only):

```
* soft  nofile  4096
* hard  nofile  65536
* soft  nproc   2047
* hard  nproc   16384
```

The `nofiles` values represent the open file limit; the `nproc` values represent the number of processes limit.

3. Save the changes, and close the file.
4. Re-login into the host computer.

Verifying IP Addresses and Host Names in DNS or Hosts File

Before you begin the installation of the Oracle software, ensure that the IP address, fully qualified host name, and the short name of the host are all registered with your DNS server. Alternatively, you can use the local `hosts` file and add an entry similar to the following:

```
IP_Address Fully_Qualified_Name Short_Name
```

For example:

```
10.229.188.205  host1.example.com  host1
```

Oracle also recommends that you use aliases to map to different IPs in different data centers in preparation for disaster recovery. You can also use these aliases to configure the listen address for some of the components.

In this guide, the abstract hostnames that are provided on the **Hardware - Host Computers** tab of the workbook (OIMHOST*n* and ADMINVHN) are used for these aliases, so the `/etc/hosts` can be similar to this example:

```
10.229.188.204 host1-vip.example.com host1-vip ADMINVHN
10.229.188.205 host1.example.com host1 OIMHOST1
10.229.188.206 host2.example.com host2 OIMHOST2
10.229.188.207 host3.example.com host3 WEBHOST1
10.229.188.208 host4.example.com host4 WEBHOST2
10.229.188.208 host5.example.com host5 OAMHOST1
10.229.188.208 host6.example.com host6 OAMHOST2
```

Enabling Unicode Support

It is recommended to enable Unicode support in your operating system so as to allow processing of characters in Unicode.

Your operating system configuration can influence the behavior of characters supported by Oracle Fusion Middleware products.

On UNIX operating systems, Oracle highly recommends that you enable Unicode support by setting the `LANG` and `LC_ALL` environment variables to a locale with the UTF-8 character set. This enables the operating system to process any character in Unicode. Oracle Identity and Access Management technologies, for example, are based on Unicode.

If the operating system is configured to use a non-UTF-8 encoding, Oracle Identity and Access Management components may function in an unexpected way. For example, a non-ASCII file name might make the file inaccessible and cause an error. Oracle does not support problems caused by operating system constraints.

Setting the DNS Settings

Configure the host to access your corporate DNS hosts. To do this, update DNS settings by updating the file `/etc/resolv.conf`.

Configuring Users and Groups

Create the following groups and user either locally or in your NIS or LDAP server. This user is the Oracle Software Owner.

The instructions below are for creating the user locally. Refer to your NIS documentation for information about creating these groups and user in your NIS server.

Groups

You must create the following groups on each node.

- `oinstall`
- `dba`

To create the groups, use the following command as root:

```
groupadd groupname
```

For example

```
groupadd -g 500 oinstall
groupadd -g 501 dba
```

User

You must create the following user on each node.

- `oracle`–The owner of the Oracle software. You may use a different name. The primary group for this account must be `oinstall`. The account must also be in the `dba` group.

> **Note:**
>
> - The group `oinstall` must have write privileges to all the file systems on shared and local storage that are used by the Oracle software.
> - Each group must have the same Group ID on every node.
> - Each user must have the same User ID on every node.
> - The user and group should exists at the NIS server due to the NFSv4 mount requirement.

To create a local user, use the following command as `root`:

```
useradd -g primary group -G optional groups -u userid username
```

For example:

```
useradd -g oinstall -G dba -u 500 oracle
```

> **Note:**
>
> To create this user in NIS, refer to your NIS documentation.

Configuring a Host to Use an NTP (time) Server

All servers in the deployment must have the same time. The best way to achieve this is to use an NTP server. To configure a host to use an NTP server:

1. Determine the name of the NTP server(s) you wish to use. For security reasons, ensure that these are inside your organization.
2. Log in to the host as the root user.
3. Edit the file `/etc/ntp.conf` to include a list of the time servers. After editing, the file appears as follows:

   ```
   server ntphost1.example.com
   server ntphost2.example.com
   ```

4. Run the following command to synchronize the system clock to the NTP server:

   ```
   /usr/sbin/ntpdate ntpserver1.example.com
   /usr/sbin/ntpdate ntpserver2.example.com
   ```

5. Start the NTP client using the following command:

   ```
   service ntpd start
   ```

6. Validate that the time is set correctly using the date command.
7. To make sure that the server always uses the NTP server to synchronize the time. Set the client to start on reboot by using the following command:

   ```
   chkconfig ntpd on
   ```

Configuring a Host to Use an NIS/YP Host

If you are using NFS Version 4, configure a directory service or an NIS (Network Information Server).

Once you have configured your NIS host, configure each compute node to use it. Before beginning, determine the host names of the NIS servers you are going to use.

1. Login to the host as root.
2. Edit the `/etc/idmapd.conf` configuration file:

   ```
   vi /etc/idmapd.conf
   ```

   Set the domain value, as in the following example:

   ```
   Domain = example.com
   ```

3. Restart the `rpcidmapd` service:

   ```
   service rpcidmapd restart
   ```

4. Update the `/etc/yp.conf` configuration file, and set the correct domain value, as in the following example:

   ```
   vi /etc/yp.conf
   ```

Add the following line:

```
domain example.com server NIS_Server_hostname_or_IP
```

Where `example.com` is the example domain and *NIS_Server_hostname_or_IP* is the host name or IP address of the NIS host. You must replace these sample values with values appropriate for your environment.

5. Set NIS domain name on the command line:

   ```
   domainname NIS_DOMAIN_NAME
   ```

   For example:

   ```
   domainname nisdomain.example.com
   ```

6. Edit the `/etc/nsswitch.conf` configuration file:

   ```
   vi /etc/nsswitch.conf
   ```

   Add `nis` to each of the following entries:

   > **Note:**
   >
   > The first value may be `compat` or `files` depending on your OS and enterprise requirements.

   ```
   passwd:     files nis
   shadow:     files nis
   group:      files nis
   automount:  files nis nisplus
   aliases:    files nis nisplus
   ```

7. Restart the `rpcidmapd` service:

   ```
   service rpcidmapd restart
   ```

8. Restart the `ypbind` service by running the following command:

   ```
   service ypbind restart
   ```

9. Check the `yp` service by running this command:

   ```
   ypwhich
   ```

10. Verify if you can access Oracle user accounts:

    ```
    ypcat passwd
    ```

11. Add `ypbind` to your boot sequence, so that it starts automatically after rebooting.

    ```
    chkconfig ypbind on
    ```

Mounting the Required Shared File Systems on Each Host

It is important to understand how to mount the shared storage to all the servers that require access.

The shared storage configured, as described in Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment, must be available on the hosts that use it.

In an enterprise deployment, it is assumed that you have a hardware storage filer, which is available and connected to each of the host computers that you have procured for the deployment.

You must mount the shared storage to all servers that require access.

Each host must have appropriate privileges set within the Network Attached Storage (NAS) or Storage Area Network (SAN) so that it can write to the shared storage.

Follow the best practices of your organization for mounting shared storage. This section provides an example of how to do this on Linux by using NFS storage.

You must create and mount shared storage locations so that OAMHOST1 and OAMHOST2 can see the same location if it is a binary installation in two separate volumes.

See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

You use the following command to mount shared storage from a NAS storage device to a Linux host. If you are using a different type of storage device or operating system, refer to your manufacturer documentation for information about how to do this.

> **Note:**
>
> The user account used to create a shared storage file system owns and has read, write, and execute privileges for those files. Other users in the operating system group can read and process the files, but they do not have write privileges.
>
> See Selecting an Installation User in the *Oracle Fusion Middleware Installation Planning Guide*.

In the following example, `nasfiler` represents the shared storage filer. Also note that these are examples only. Typically, the mounting of these shared storage locations should be done by using the `/etc/fstabs` file on UNIX systems, so that the mounting of these devices survives a reboot. Refer to your operating system documentation for more information.

To mount the shared storage on Linux:

1. Create the mount directories on OAMHOST1, as described in Summary of the Shared Storage Volumes in an Enterprise Deployment, and then mount the shared storage. For example:

   ```
   mount-tnfs nasfiler:VOL1/oracle/products/ /u01/oracle/products/
   ```

2. Repeat the procedure on OAMHOST2 using VOL2.

Validating the Shared Storage Configuration

Ensure that you can read and write files to the newly mounted directories by creating a test file in the shared storage location that you just configured.

For example:

```
$ cd newly mounted directory
$ touch testfile
```

Verify that the owner and permissions are correct:

```
$ ls -l testfile
```

Then remove the file:

```
$ rm testfile
```

> **Note:**
>
> The shared storage can be a NAS or SAN device. The following example illustrates creating storage for a NAS device from OAMHOST1. The options may differ depending on the specific storage device.
>
> ```
> mount -t nfs -o rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsize=32768
> nasfiler:VOL1/Oracle/u01/oracle
> ```
>
> Contact your storage vendor and machine administrator to learn about the appropriate options for your environment.

Enabling the Required Virtual IP Addresses on Each Host

You must enable the required virtual IP addresses on each host in order to prepare the host for the enterprise deployment.

To prepare each host for the enterprise deployment, you must enable the virtual IP (VIP) addresses that are described in Reserving the Required IP Addresses for an Enterprise Deployment.

It is assumed that you have already reserved the VIP addresses and host names and that they have been enabled by your network administrator. You can then enable the VIPs on the appropriate host.

Note that the virtual IP addresses used for the enterprise topology are not persisted because they are managed by Whole Server Migration (for selected Managed Servers and clusters) or by manual failover (for the Administration Server).

Starting with Oracle Enterprise Linux 6, the "`ifconfig`" command is deprecated and is replaced with the "`ip`" command.

To enable the VIP addresses on each host, run the following commands as `root`:

1. Determine the CIDR notation of the netmask. Each Netmask has a CIDR notation. For example, 255.255.240.0 has a CIDR of 20.

If the netmask you are adding is the same as the interface, the fastest way to determine this is to examine the existing IP address that are assigned to the network card. You can do this by using the following command:

```
ip addr show dev eth0
```

Sample output:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen
1000
link/ether 00:21:f6:03:85:9f brd ff:ff:ff:ff:ff:ff
int 192.168.20.1/20 brd 10.248.11.255 scope global eth0
```

In this example, the CIDR value is the value after the forward slash (/), which is, 20. If you are unsure of the CIDR value, contact your network administrator.

2. Configure the additional IP address on the appropriate network interface card with an appropriately suffixed label using the following command:

```
ip addr add VIP/CIDR dev nic# label nic#:n
```

> **Note:**
>
> For each VIP/VHN that you need to add, increment the :n suffix starting with :1

Example: For VIP IP of 192.168.20.3, netmask: 255.255.240.0 (CIDR: 20), and the eth0 NIC:

```
ip addr add 192.168.20.3/20 dev eth0 label eth0:1
```

3. For each of the virtual IP addresses that you define, update the ARP caches by using the following command:

```
arping -b -A -c 3 -I eth0 192.168.20.3
```

11
Preparing the Database for an Enterprise Deployment

Preparing the database for an enterprise deployment involves ensuring that the database meets specific requirements, creating database services, using SecureFiles for large objects in the database, and creating database backup strategies.

This chapter provides information about the database requirements, creating database services, and about the database backup strategies.

- About Preparing the Database for an Enterprise Deployment
  You have to configure a supported database as part of an Oracle Fusion Middleware enterprise deployment. Most Oracle Fusion Middleware products require a specific set of schemas that must be installed in a supported database. The schemas are installed by using the Oracle Fusion Middleware Repository Creation Utility (RCU).
- About Database Requirements
  Before you configure the enterprise deployment topology, you have to verify that the database meets the requirements described in the following sections.
- Applying the Database Patches
  Ensure that you apply all the necessary database patches. The patches are required only if you are using Oracle Identity Governance.
- Creating Database Services
  When multiple Oracle Fusion Middleware products are sharing the same database, each product should be configured to connect to a separate, dedicated database service. This service should be different from the default database service.
- Using SecureFiles for Large Objects (LOBs) in an Oracle Database
  SecureFiles is a new LOB storage architecture introduced in Oracle Database 11g Release 1. It is recommended to use SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas.
- About Database Backup Strategies
  Performing a database backup at key points in the installation and configuration of an enterprise deployment enables you to recover quickly from any issue that might occur in the later configuration steps.

About Preparing the Database for an Enterprise Deployment

You have to configure a supported database as part of an Oracle Fusion Middleware enterprise deployment. Most Oracle Fusion Middleware products require a specific set of schemas that must be installed in a supported database. The schemas are installed by using the Oracle Fusion Middleware Repository Creation Utility (RCU).

In an enterprise deployment, Oracle recommends a highly available Real Application Clusters (Oracle RAC) database for the Oracle Fusion Middleware product schemas.

About Database Requirements

Before you configure the enterprise deployment topology, you have to verify that the database meets the requirements described in the following sections.

- Supported Database Versions
- Additional Database Software Requirements
- Databases Required
- Minimum Initialization Parameters

Supported Database Versions

Use the following information to verify what databases are supported by each Oracle Fusion Middleware release and which version of the Oracle database you are currently running:

- For a list of all certified databases, refer to *Oracle Fusion Middleware Supported System Configurations*.
- To check the release of your database, query the `PRODUCT_COMPONENT_VERSION` view:

```
SQL> SELECT VERSION FROM SYS.PRODUCT_COMPONENT_VERSION WHERE
        PRODUCT LIKE 'Oracle%';
```

Oracle Fusion Middleware requires that the database supports the AL32UTF8 character set. Check the database documentation for information on choosing a character set for the database.

For enterprise deployments, Oracle recommends that you use GridLink data sources to connect to Oracle RAC databases.

> **Note:**
>
> For more information about using GridLink data sources and SCAN, see Using Active GridLink Data Sources in *Administering JDBC Data Sources for Oracle WebLogic Server*.
>
> Use of Active GridLink has specific licensing requirements, including a valid WebLogic Suite license. See Oracle WebLogic Server data sheet.

Additional Database Software Requirements

In the enterprise topology, there are two database host computers in the data tier that host the two instances of the RAC database. These hosts are referred to as DBHOST1 and DBHOST2.

Before you install or configure the enterprise topology, you must ensure that the following software is installed and available on DBHOST1 and DBHOST2:

- **Oracle Clusterware**

  See Installing Oracle Grid Infrastructure for a Cluster in *Oracle Grid Infrastructure Installation Guide for Linux*.
- **Oracle Real Application Clusters**

See Installing Oracle RAC and Oracle RAC One Node in *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

- **Time synchronization between Oracle RAC database instances**

  The clocks of the database instances must be in sync if they are used by servers in a Fusion Middleware cluster configured with server migration.

- **Automatic Storage Management** (optional)

  See Introducing Oracle Automatic Storage Management in *Oracle Automatic Storage Management Administrator's Guide*.

General Database Characteristics

- Character Set – The character set must be unicode compliant. For example: AL32UTF8.
- Database Options – The following database options must be installed into the database:
  - Oracle JVM
  - Oracle Text
- Database Views – The following database view must be created on the database:
  - XAVIEWS
- Database Packages – The following database package must exist in the database:
  - DBMS_SHARED_POOL
- Transparent Data Encryption - This is required by the Oracle Privileged Account Manager.

Databases Required

For Oracle Identity and Access Management, a number of separate databases are recommended. Table 11-1 provides a summary of these databases. Which database or databases you use depends on the topology that you are implementing.

For this release of Oracle Identity and Access Management, you must use a separate RCU schema prefix for each domain. This allows different products to use a the same or different databases if required.

If you are planning on creating a Multi-Datacenter, you should use separate databases for Access and Governance. This allows different replication mechanisms to be used for each.

Table 11-1 Mapping between Databases and Schemas

Database Names	Database Hosts	Scan Address	Service Name	RCU Prefix	Schemas in Database
IADDB	DBHOST1 DBHOST2	*DBSCAN*	`iadedg.example.com`	EDGIAD	OAM, IAU, MDS, OPSS
IGDDB	DBHOST1 DBHOST2	*DBSCAN*	`igdedg.example.com`	EDGIGD	OIM, SOAINFRA, MDS, OPSS, ORASDPM, BI, ODS

> **Note:**
>
> - Databases can be combined if you are not planning on creating a multi-datacenter deployment.
> - Databases can be pluggable.

Minimum Initialization Parameters

The databases must have the following minimum initialization parameters defined:

Table 11-2 Minimum Initialization Parameters for Oracle Databases

Parameter	Small Value	Medium Value	Large Value
aq_tm_processes	1	1	1
dml_locks	200	200	200
job_queue_processes	12	12	12
open_cursors	1600	1600	1600
session_max_open_files	50	50	50
sessions	4000	4000	4000
processes	5000	5000	5000
sga_target	28G	58G	118G
pga_aggregate_target	7G	14G	29G
sga_max_size	8G	8G	8G
session_cached_cursors	800	800	800
db_keep_cache_size	800M	800M	800M
cursor_sharing	FORCE	FORCE	FORCE
query_rewrite_integrity	TRUSTED	TRUSTED	TRUSTED
query_rewrite_enabled	TRUE	TRUE	TRUE
max_dispatchers	0	0	0
max_shared_servers	0	0	0
_active_session_legacy_behavior	TRUE	TRUE	TRUE
disk_asynch_io	FALSE	FALSE	FALSE
_b_tree_bitmap_plans	FALSE	FALSE	FALSE
parallel_max_servers	1	1	1
shared_servers	0	0	0
db_securefile	ALWAYS	ALWAYS	ALWAYS
plsql_code_type	NATIVE	NATIVE	NATIVE
_active_session_legacy_behavior	TRUE	TRUE	TRUE

Oracle recommends you to set these parameters in the database configuration assistant when creating the database. If not, you can adjust them after creating the database, by using the `alter system` database command. For example:

```
sqlplus / as sysdba
alter system set aq_tm_processes=1 scope=spfile;
```

After making changes in the `spfile`, restart the database. For example

```
srvctl stop database -d iaddb
srvctl start database -d iaddb
```

> **Note:**
>
> For guidelines on setting up optimum parameters for the Database, see Tuning Database Parameters in *Tuning Performance*.

Applying the Database Patches

Ensure that you apply all the necessary database patches. The patches are required only if you are using Oracle Identity Governance.

Include the database patches from Oracle Text Mandatory Patches.

Also, ensure that you include the latest patch set bundles for the release of database that you are using.

Creating Database Services

When multiple Oracle Fusion Middleware products are sharing the same database, each product should be configured to connect to a separate, dedicated database service. This service should be different from the default database service.

A different service name enables you to create role-based database services for Disaster Recovery and Multi-Datacenter topologies.

> **Note:**
>
> The instructions in this section are for the Oracle Database 12*c* (12.1) release. If you are using another supported database, refer to the appropriate documentation library for more up-to-date and release-specific information.

Beginning with Data Guard 11*g* Release 2, you can automatically control the startup of database services on primary and standby database by assigning a database role to each service. This service is in addition to the default service created when the database was commissioned. A role based database service will automatically start upon database startup if the management policy of the service is AUTOMATIC and if one of the roles assigned to that service matches the current role of the database; for example, if the database is running as a primary.

Creating a database service in this way means that, the service is started whenever the database with the role `primary` is started. The service will move between sites as the underlying databases roles are moved through switchover or failover.

If you are planning to use a standard disaster recovery solution as described in Disaster Recovery Guide, then each database service should be defined as a Role Based Database service.

If you are planning on using a multi-datacenter deployment, then the database service created for the Oracle Identity Governance (IGDDB) database should be a role based service.

For more information about connecting to Oracle databases using services, see Overview of Using Dynamic Database Services to Connect to Oracle Databases in *Real Application Clusters Administration and Deployment Guide*.

In addition, the database service should be different from the default database service. For complete instructions on creating and managing database services for an Oracle Database 12*c* database, see Overview of Automatic Workload Management with Dynamic Database Services in *Real Application Clusters Administration and Deployment Guide*.

Runtime connection load balancing requires configuring Oracle RAC Load Balancing Advisory with service-level goals for each service for which load balancing is enabled.

You can configure the Oracle RAC Load Balancing Advisory for `SERVICE_TIME` or `THROUGHPUT`. Set the connection load-balancing goal to **SHORT**.

You create and modify Oracle Database services by using the `srvctl` utility.

To create and modify a database service:

1. Add the service to the database and assign it to the instances by using `srvctl`:

   ```
   srvctl add service -db iaddb -service iamedg.example.com -preferred iaddb1,iaddb2
   ```

   If you use PDB, change the command to the following:

   ```
   srvctl add service -db iamdb1 -pdb iadpdb -service iadedg.example.com -
   preferred iamdb11, iamdb12
   ```

   > **Note:**
   >
   > For the Service Name of the Oracle RAC database, use lowercase letters, followed by the domain name. For example: `iamedg.example.com`.

2. Start the service:

   ```
   srvctl start service -db iaddb -service iamedg.example.com
   ```

   > **Note:**
   >
   > For complete instructions on creating and managing database services with SRVCTL, see Creating Services with SRVCTL in the *Real Application Clusters Administration and Deployment Guide*.

3. Modify the service so that it uses the Load Balancing Advisory and the appropriate service-level goals for runtime connection load balancing.

Use the following resources in the Oracle Database 12*c Real Application Clusters Administration and Deployment Guide* to set the SERVICE_TIME and THROUGHPUT service-level goals:

- Overview of the Load Balancing Advisory
- Configuring Your Environment to Use the Load Balancing Advisory

For example:

Check the default configuration of the service by using this command:

```
srvctl config service -db iaddb -service iamedg.example.com
```

Several parameters are shown. Check the following parameters:

- Connection Load Balancing Goal: Long
- Runtime Load Balancing Goal: NONE

You can modify these parameters by using the following command:

```
srvctl modify service -db iaddb -service iamedg.example.com -rlbgoal
SERVICE_TIME -clbgoal SHORT
```

4. Restart the service:

```
srvctl stop service -db iamdb -service iamedg.example.com
srvctl start service -db iamdb -service iamedg.example.com
```

5. Verify the change in the configuration:

```
srvctl config service -db iamdb -service iamedg.example.com


Runtime Load Balancing Goal: SERVICE_TIME
  Service name: iamedg.example.com
  Service is enabled
  Server pool: iamdb_iamedg.example.com
  ...
  Connection Load Balancing Goal: SHORT
  Runtime Load Balancing Goal: SERVICE_TIME
  ...
```

Using SecureFiles for Large Objects (LOBs) in an Oracle Database

SecureFiles is a new LOB storage architecture introduced in Oracle Database 11g Release 1. It is recommended to use SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas.

Beginning with Oracle Database 11g Release 1, Oracle introduced SecureFiles, a new LOB storage architecture. Oracle recommends that you use SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas. See Using Oracle SecureFiles LOBs in the *Oracle Database SecureFiles and Large Objects Developer's Guide*.

In Oracle 12*c* Databases, the default setting for using SecureFiles is `PREFERRED` . This means that the database attempts to create a SecureFiles LOB unless a BasicFiles LOB is explicitly specified for the LOB or the parent LOB (if the LOB is in a partition or sub-partition). The Oracle Fusion Middleware schemas do not explicitly specify BasicFiles, which means that Oracle Fusion Middleware LOBs defaults to SecureFiles when installed in an Oracle 12*c* database.

For Oracle 11*g* databases, the `db_securefile` system parameter controls the SecureFiles usage policy. This parameter can be modified dynamically. The following options can be used for using SecureFiles:

- `PERMITTED`: Allows SecureFiles to be created (This is the default setting for `db_securefile`. The default storage method uses BasicFiles).
- `FORCE`: Creates all (new) LOBs as SecureFiles.
- `ALWAYS`: Tries to create LOBs as SecureFiles, but falls back to BasicFiles if not possible (if ASSM is disabled).

Other values for the `db_securefile` parameter are:

- `IGNORE`: Ignore attempts to create SecureFiles.
- `NEVER`: Disallow new SecureFiles creations.

For Oracle 11*g* Databases, Oracle recommends that you set the `db_securefile` parameter to `FORCE` before you create the Oracle Fusion Middleware schemas with the Repository Creation Utility (RCU).

Note that the SecureFiles segments require tablespaces managed with automatic segment space management (ASSM). This means that LOB creation on SecureFiles will fail if ASSM is not enabled. However, the Oracle Fusion Middleware tablespaces are created by default with ASSM enabled. As a result, with the default configuration, nothing needs to be changed to enable SecureFiles for the Oracle Fusion Middleware schemas.

About Database Backup Strategies

Performing a database backup at key points in the installation and configuration of an enterprise deployment enables you to recover quickly from any issue that might occur in the later configuration steps.

At key points in the installation and configuration of an enterprise deployment, this guide recommends that you back up your current environment. For example, after you install the product software and create the schemas for a particular Oracle Fusion Middleware product, you should perform a database backup. Performing a backup allows you to perform a quick recovery from any issue that might occur in the later configuration steps.

You can choose to use your own backup strategy for the database, or you can simply make a backup by using operating system tools or RMAN for this purpose.

Oracle recommends that you use Oracle Recovery Manager for the database, particularly if the database was created using Oracle Automatic Storage Management. If possible, you can also perform a cold backup by using operating system tools such as tar.

Part III

Configuring the Enterprise Deployment

You have to perform a series of steps to configure your enterprise deployment. For example, creating the initial infrastructure domain, which you can use as the starting point for the deployment. This section provides instructions to complete each of these steps, in detail.

Part III contains the following chapters:

- Configuring Oracle LDAP for an Enterprise Deployment
  Follow these instructions if you are creating a new Oracle LDAP directory (Oracle Unified Directory (OUD)).
- Creating Infrastructure for Oracle Access Management
- Creating Infrastructure for Oracle Identity Governance
- Configuring Oracle HTTP Server for an Enterprise Deployment
  For an enterprise deployment, Oracle HTTP Server must be installed on each of the web tier hosts and configured as Oracle HTTP standalone domains on each host.
- Configuring Oracle Access Management
  You need to perform certain tasks in order to extend the enterprise deployment domain with the Oracle Access Management. This includes installing the Oracle Identity and Access Management, extending the domain for Oracle Access Management and completing post-configuration and verification tasks.
- Configuring Oracle Identity Governance
  Configuration of Oracle Identity Governance (OIG) comprises a series of steps, including integrating OIG with Oracle SOA suite, configuring the web tier, integrating OAM and OIG, configuring OIG workflow notifications, and so on. In the end, you back up the configuration.
- Configuring Multi-Data Center
  Multi-Data Centers (MDC) help you distribute load as well as address disaster recovery. This chapter provides detailed instructions to help you configure multi- data centers for your enterprise deployment.

12

Configuring Oracle LDAP for an Enterprise Deployment

Follow these instructions if you are creating a new Oracle LDAP directory (Oracle Unified Directory (OUD)).

This chapter includes the following topics:

- Configuring Oracle Unified Directory
  Install and configure Oracle Unified Directory (OUD). In an enterprise deployment, each OUD instance is configured on a separate host. OUD is not installed into a domain.
- Configuring Oracle HTTP Server for Oracle Unified Directory Services Manager
  If you want to access the Oracle Unified Directory Services Manager (OUDSM) console through Oracle Web Servers, then you must add the necessary entry to one of your administrative virtual hosts.
- Preparing an Existing LDAP Directory
  Before you can use an LDAP directory with Oracle Identity and Access Management, it must be extended with object classes required by Oracle Access Manager.

Configuring Oracle Unified Directory

Install and configure Oracle Unified Directory (OUD). In an enterprise deployment, each OUD instance is configured on a separate host. OUD is not installed into a domain.

- Variables Used When Configuring Oracle Unified Directory
  The procedures for installing and configuring Oracle Unified Directory reference use a series of variables that you can replace with the actual values used in your environment.
- Installing a Supported JDK
- Installing Oracle Unified Directory
  You can install Oracle Unified Directory by using an interactive graphical wizard provided by the Oracle Universal Installer.
- Configuring the Oracle Unified Directory Instances
- Installing and Configuring Oracle Unified Directory Service Manager
  Oracle Unified Directory Service Manager (OUDSM) is a Graphical User Interface (GUI) tool used to manage Oracle Unified Directory.

Variables Used When Configuring Oracle Unified Directory

The procedures for installing and configuring Oracle Unified Directory reference use a series of variables that you can replace with the actual values used in your environment.

The following directory location variables are used in these procedures:

- *DIR_ORACLE_HOME*
- *OUD_ORACLE_INSTANCE*
- *OUD_REPLICATION_PORT*

- *OUD_ADMIN_PORT*
- *LDAP_PORT*
- *LDAP_SSL_PORT*
- *LDAP_ADMIN_PORT*
- *JAVA_HOME*
- *INSTANCE_NAME*
- *PRIVATE_CONFIG_DIR*
- *WEB_DOMAIN_HOME*
- *OHS_DOMAIN_HOME*
- *IDSTORE_HOST*
- *IDSTORE_PORT*
- *IDSTORE_DIRECTORYTYPE*
- *IDSTORE_BINDDN*
- *IDSTORE_SEARCHBASE*
- *IDSTORE_LOGINATTRIBUTE*
- *IDSTORE_USERSEARCHBASE*
- *IDSTORE_GROUPSEARCHBASE*
- *IDSTORE_SYSTEMIDBASE*
- *IDSTORE_USERNAMEATTRIBUTE*
- *IDSTORE_LOGIN_ATTRIBUTE*
- *IDSTORE_ADMIN_PORT*
- *IDSTORE_KEYSTORE_FILE*
- *IDSTORE_KEYSTORE_PASSWORD*
- *IDSTORE_NEW_SETUP*
- *IDSTORE_OAMADMINUSER*
- *IDSTORE_OAMSOFTWAREUSER*
- *OAM11G_IDSTORE_ROLE_SECURITY_ADMIN*
- *OAM11G_SERVER_LOGIN_ATTRIBUTE*
- *IDSTORE_WLSADMINUSER*
- *IDSTORE_WLSADMINGROUP*
- *IAD_ORACLE_HOME*
- *IGD_ORACLE_HOME*
- *ORACLE_HOME*

Installing a Supported JDK

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system.

- Locating and Downloading the JDK Software
- Installing the JDK Software
  Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

Locating and Downloading the JDK Software

To find a certified JDK, see the certification document for your release on the Oracle Fusion Middleware Supported System Configurations page.

After you identify the Oracle JDK for the current Oracle Fusion Middleware release, you can download an Oracle JDK from the following location on Oracle Technology Network:

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the following locations:

- On the shared storage device, where it will be accessible from each of the application tier host computers, install the JDK in the location specified in File System and Directory Variables Used in This Guide.
- On the local storage device for each of the Web tier host computers. The Web tier host computers, which reside in the DMZ, do not necessarily have access to the shared storage on the application tier.
- On the local storage device for each of the directory tier host computers, in case of the directory hosts not utilizing the shared storage.

For more information about the recommended location for the JDK software, see Understanding the Recommended Directory Structure for an Enterprise Deployment.

To install JDK 1.8.0_211:

1. Change directory to the location where you downloaded the JDK archive file.

   ```
   cd download_dir
   ```

2. Unpack the archive into the JDK home directory, and then run the following commands:

   ```
   tar -xzvf jdk-8u201-linux-x64.tar.gz
   ```

   Note that the JDK version listed here was accurate at the time this document was published. For the latest supported JDK, see the *Oracle Fusion Middleware System Requirements and Specifications* for the current Oracle Fusion Middleware release.

3. Move the JDK directory to the recommended location in the directory structure.

   For example:

   ```
   mv ./jdk1.8.0_211 /u01/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

For example:

```
export JAVA_HOME=/u01/oracle/products/jdk
export PATH=$JAVA_HOME/bin:$PATH
```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

```
java -verison
```

The Java version in the output should be displayed as “1.8.0_211”.

Installing Oracle Unified Directory

You can install Oracle Unified Directory by using an interactive graphical wizard provided by the Oracle Universal Installer.

- Starting the Oracle Unified Directory Installer
- Navigating the Oracle Unified Directory Installation Screens
- Installing the Software on Other Host Computers
- Verifying the Installation

Starting the Oracle Unified Directory Installer

To start the installation program:

1. Log in to LDAPHOST1.
2. Go to the directory in which you downloaded the installer.
3. Run the following Java command to launch the installation wizard:
   - On Linux

     ```
     JAVA_HOME/bin/java -d64 -jar fmw_12.2.1.4.0_oud_generic.jar
     ```

   Replace the JDK location in the above command with the actual JDK location on your system. For information about downloading the software and locating the actual installer file name for your product, see Identifying and Obtaining Software Distributions for an Enterprise Deployment.

Navigating the Oracle Unified Directory Installation Screens

The following table describes how to use the installer screens to install Oracle Unified Directory.

If you need additional help with any of the installation screens, click the screen name.

Screen	Description
Welcome	This screen introduces you to the product installer. Click **Next**.
Auto Updates	Select whether or not you want to receive automatic updates for this product.
Installation Location	For the purposes of this enterprise deployment, enter the value of the *DIR_ORACLE_HOME* variable listed in Table 8-2. Note that run-time processes cannot write to this directory.

Screen	Description
Installation Type	Use this screen to select the type of installation and as a consequence, the products and feature sets you want to install. If you plan to manage OUD through WebLogic server or OUDSM, select **Collocated Oracle Unified Directory Server (Managed through WebLogic server)**. **Note:** If you select Collocated mode, you must also install Oracle Fusion Middleware Infrastructure. See Installing the Oracle Fusion Middleware Infrastructure. If you plan to manage OUD independently of WebLogic server, select **Standalone Oracle Unified Directory Server (Managed independently of WebLogic server)**. Click **Next**.
Prerequisite Checks	The installer analyzes the host computer to ensure that the prerequisites are fulfilled. The results of the prerequisite checks are displayed on this screen. If a prerequisite check fails, an error or warning message is displayed. • Fix the error and click **Rerun**. • To ignore the error or warning and continue with the installation, click **Skip**. • To stop the prerequisite checking process, click **Stop**. Click **Next** to continue.
Installation Summary	This screen displays the Oracle home directory that you specified earlier. It also indicates the amount of disk space that will be used for the installation and the free space available. Review information on this screen. To save the settings specified so far in the installation wizard in a text file (called a *response* file), click **Save**. If necessary, you can use the response file to perform the same installation from the command line. Click **Install** to begin the installation. For more information about silent or command line installation, see "Using the Oracle Universal Installer in Silent Mode" in Installing Software with the Oracle Universal Installer.

Screen	Description
Installation Progress	This screen shows the progress and status of the installation process. If you want to cancel the installation, click **Cancel**. The files that were copied to your system before you canceled the installation will remain on the system; you should remove them manually. Click **Next** to continue.
Installation Complete	Click **Finish**.

Installing the Software on Other Host Computers

If you have configured a separate shared storage volume or partition for LDAPHOST2 , then you must also install the software on LDAPHOST2. For more information, see Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

Note that the location where you install the Oracle home (which contains the software binaries) varies, depending upon the host. To identify the proper location for your Oracle home directories, refer to the guidelines in File System and Directory Variables Used in This Guide.

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- Reviewing the Installation Log Files
- Checking the Directory Structure
  After you install the Oracle Unified Directory and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options you selected during the installation.
- Viewing the Contents of Your Oracle Home

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see Understanding Installation Log Files in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

After you install the Oracle Unified Directory and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options you selected during the installation.

To check the directory structure:

1. Change to the *DIR_ORACLE_HOME* directory where you installed the Oracle Unified Directory.
2. Enter the following command:

   ```
   ls --format=single-column
   ```

   If you installed using the colocated method, the directory structure on your system must match the structure shown in the following example:

```
addons
bat
bin
common
config
lib
libForUpgrade
oud-proxy-setup
oud-proxy-setup.bat
oud-replication-gateway-setup
oud-replication-gateway-setup.bat
oud-setup
oud-setup.bat
plugins
snmp
winlib
```

If you installed using the standalone method, then the directory structure should match the structure shown below:

```
cfgtoollogs
inventory
OPatch
oracle_common
oraInst.loc
oud
oui
wlserver
```

See What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See Viewing the contents of an Oracle home in *Installing Software with the Oracle Universal Installer*.

Configuring the Oracle Unified Directory Instances

Follow these steps to configure Oracle Unified Directory (OUD) components in the directory tier on LDAPHOST1 and LDAPHOST2. During the configuration you will also configure Oracle Unified Directory replication servers.

The following are the two option available when you install Oracle Unified Directory:

- Standalone mode: Choose this option if you wish to manage OUD via command line tools.
- Co-located mode: Choose this option to associate Oracle Unified directory with a domain. If you choose to associate it with a domain, you have the option to manage OUD using Oracle Unified Directory Service Manager. If you wish to use OUDSM, you must choose to install Oracle Unified Directory in co-located mode.

This section contains the following topics:

- Configuring Oracle Unified Directory on LDAPHOST1
- Validating Oracle Unified Directory on LDAPHOST1
- Configuring Oracle Unified Directory Instance on LDAPHOST2
- Validating Oracle Unified Directory on LDAPHOST2

Configuring Oracle Unified Directory on LDAPHOST1

Ensure that ports 1389, 1636, 4444, and 8989 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.

```
netstat -an | grep "1389"
```

If the ports are in use (that is, if the command returns output identifying either port), you must free the port.

Remove the entries for ports 1389, 1636, 4444, and 8989 in the `/etc/services` file and restart the services or restart the computer.

Set the environment variable JAVA_HOME

Change Directory to *DIR_ORACLE_HOME*/oud.

Set the environment variable INSTANCE_NAME to `../../admin/oud1`. For example:

```
export INSTANCE_NAME=../../../../u02/private/oracle/config/instances/oud1
```

Note the tool creates the instance home relative to the *DIR_ORACLE_HOME*, so you must include previous directories to get the instance created in *PRIVATE_CONFIG_DIR*/instances.

Start the Oracle Unified Directory configuration assistant by executing the command:

```
./oud-setup
```

The following table describes how to use the configuration assistant screens to configure Oracle Unified Director.

Screen	Description
Welcome	This screen introduces you to the product configuration assistant. Click **Next**.

Screen	Description
Server Administration Settings	Enter the following details of the server: • **Instance Path**: Enter the location of the OUD configuration files (*OUD_INSTANCE_HOME*). • **Host Name**: Enter the name of the host where Oracle Unified Directory is running. For example: `LDAPHOST1.example.com` • **Administration Port(s)**: The value of this field determines how you are going to administer OUD. The following are the optional values: – **Enable Administration only by LDAP**: Enter the LDAP port that will be used for administration traffic. The default LDAP administration port is `4444`. – **Enable Administration by LDAP and HTTP**: Enter the LDAP and HTTP ports that will be used for administration traffic. The default administration ports are `4444` for LDAP and `8444` for HTTP. – **Enable Administration by HTTP**: Enter the HTTP port that will be used for administration traffic. The default HTTP administration port is `8444`. • **LDAP Port**: Enter the port that you wish to use for administering OUD via LDAP. • **HTTP Port**: Enter the port that you wish to use for administering OUD via HTTP: 8444 (OUD_ADMIN_PORT). • **Root User DN**: Enter an administrative user. For example, cn=oudadmin. • **Password**: Enter the password you wish to assign to the ouadmin user. • **Password (Confirm)**: Repeat the password. Click **Next**.

Screen	Description
Ports	Enter the following details: **LDAP** • **Enable**: Select if you wish to enable non SSL communications with OUD. • **on Port**: Select the Port you wish to use (*LDAP_PORT*). • **Enable Start TLS for LDAP**: Select Enable **StartTLS for LDAP** to specify that the LDAP connection handler should allow clients to use the StartTLS extended operation to initiate secure communication over an otherwise insecure connection. **LDAPS** • **Enable**: Select if you wish to enable SSL communications with OUD. • **on Port**: Select the Port you wish to use (*LDAP_SSL_PORT*). If you select this option, you must provide the SSL certificate information below. **Certificate** You can use the existing certificate or generate a self signed certificate. Self signed certificates are not recommended for production deployments. • **Generate self signed certificate**: Select this if you wish OUD to generate its own certificate. • **Use an existing certificate**: Select this if you are using an existing certificate. Select the type of certificate, the location of the Keystore, and the Keystore pin.
Topology Options	Enter the following details: • **This will server will be part of a replication topology**: Select this. • **Replication Port**: Enter the replication port. For example: `8989 (`*`OUD_REPLICATION_PORT`*`)` • **Configure As Secure**: Select this if you wish the replication traffic to be encrypted. • There is already a server in the topology. Leave it unselected. Click **Next**.
Directory Data	Enter the following details: • **Directory Base DN**: `dc=example,dc=com` • **Directory Data**: Only create base entry. Click **Next**.
Oracle Components Integration	If you are planning to use the directory for integrating with other directories, Select **Enable for DIP**. If you are planning on using the directory for E Business Suite or for Oracle database name resolution, select **Enable for EBS (E-Business Suite), Database Net Services and DIP**. Click **Next**. If you are planning to use the directory for Enterprise User Security, select **Enable fo EUS (Enterprise User Security), EBS, Database Net Services and DIP**.

Screen	Description
Server Tuning	You have the option of allocating specific resources to the OUD instance. You can choose to accept the default of change the resource allocations based on your deployment. If this server is used only for OUD as in a distributed deployment then be sure to check the box Dedicated Machine for OUD. Click **Next**.
Review	Verify that the information displayed is correct. If you wish the OUD server to be started after configuration, ensure that you select the option **Start the server**.
Finished	Click **Close**.

Validating Oracle Unified Directory on LDAPHOST1

After configuration, you can validate that Oracle Unified Directory is working by performing a simple search using the following command:

```
OUD_ORACLE_INSTANCE/OUD/bin/ldapsearch -h LDAPHOST1.example.com -p 1389 -D
cn=oudadmin -b "" -s base "(objectclass=*)" supportedControl
```

If Oracle Unified Directory is working correctly, you will see a list of `supportedControl` entries returned.

If you have enabled SSL on the directory, you can test it using the command:

```
OUD_ORACLE_INSTANCE/OUD/bin/ldapsearch -h LDAPHOST1.example.com -p 1636 --
useSSL -D cn=oudadmin -b "" -s base "(objectclass=*)" supportedControl
```

Configuring Oracle Unified Directory Instance on LDAPHOST2

Ensure that ports 1389, 1636, 4444, and 8989 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.

```
netstat -an | grep "1389"
```

If the ports are in use (that is, if the command returns output identifying either port), you must free the port.

Remove the entries for ports `1389`, `1636`, `4444`, and `8989` in the `/etc/services` file and restart the services or restart the computer.

Set the environment variable `JAVA_HOME` to *`JAVA_HOME`*.

Change Directory to *`DIR_ORACLE_HOME`*`/oud`

Set the environment variable `INSTANCE_NAME` to `../../admin/oud2`.

For example:

```
export INSTANCE_NAME=../../../../u02/private/oracle/config/instances/oud2
```

Note the tool creates the instance home relative to the *DIR_ORACLE_HOME*, so you must include previous directories to get the instance created in *PRIVATE_CONFIG_DIR*/instances.

Start the Oracle Unified Directory configuration assistant by executing the command:

```
./oud-setup
```

The following table describes how to use the configuration assistant screens to configure Oracle Unified Director.

Screen	Description
Welcome	This screen introduces you to the product configuration assistant. Click **Next**.
Server Administration Settings	Enter the following details of the server: • **Instance Path**: Enter the location of the OUD configuration files (*OUD_INSTANCE_HOME*). • **Host Name**: Enter the name of the host where Oracle Unified Directory is running. For example: `LDAPHOST2.example.com` • **Administration Port(s)**: The value of this field determines how you are going to administer OUD. The following are the optional values: – **Enable Administration only by LDAP**: Enter the LDAP port that will be used for administration traffic. – **Enable Administration by LDAP and HTTP**: Enter the LDAP and HTTP ports that will be used for administration traffic. – **Enable Administration by HTTP**: Enter the HTTP port that will be used for administration traffic. • **LDAP Port**: Enter the port that you wish to use for administering OUD via LDAP. • **HTTP Port**: Enter the port that you wish to use for administering OUD via HTTP. For example, `8444` (*OUD_ADMIN_PORT*). • **Root User DN**: Enter an administrative user. For example, `cn=oudadmin`. • **Password**: Enter the password you wish to assign to the ouadmin user. • **Password (Confirm)**: Repeat the password. Click **Next**.

Screen	Description
Ports	Enter the following details: **LDAP** - **Enable**: Select if you wish to enable non SSL communications with OUD. - **on Port**: Select the Port you wish to use (`LDAP_PORT`). - **Enable Start TLS for LDAP**: Select **Enable StartTLS for LDAP** to specify that the LDAP connection handler should allow clients to use the StartTLS extended operation to initiate secure communication over an otherwise insecure connection. **LDAPS** - **Enable**: Select if you wish to enable SSL communications with OUD. - **on Port**: Select the Port you wish to use (`LDAP_SSL_PORT`). If you select this option, you must provide the SSL certificate information below. **Certificate** You can use the existing certificate or generate a self signed certificate. Self signed certificates are not recommended for production deployments. - **Generate self signed certificate**: Select this if you wish OUD to generate its own certificate. - **Use an existing certificate**: Select this if you are using an existing certificate. Select the type of certificate, the location of the Keystore, and the Keystore pin.
Topology Options	Enter the following details: - **This will server will be part of a replication topology**: Select this. - **Replication Port**: Enter the replication port. For example: `8989 (OUD_REPLICATION_PORT)` - **Configure As Secure**: Select this if you wish the replication traffic to be encrypted. - There is already a server in the topology selected. Enter the following: – **Host Name**: Name of the existing Oracle Unified Directory server host. For example, LDAPHOST1.example.com – **Administrator Connector Port**: `4444` (`LDAP_ADMIN_PORT`) – **Admin User**: Name of the Oracle Unified Directory admin user on LDAPHOST1. For example, `cn=oudadmin` – **Admin Password**: Administrator password Click **Next**. If you see a **Certificate Not Trusted** dialogue, it is because you are using self signed certificates. Click **Accept Permanently**. For more information, see Setting Up Replication During Installation.

Screen	Description
Create Global Administrator	Enter the following details: • **Global Administrator ID**: Enter the name of an account you want to use for managing Oracle Unified Directory replication. For example: `oudmanager` • **Global Administrator Password / Confirmation**: Enter a password for this account. Click **Next**.
Data Replication	Select **dc=example,dc=com**. Click **Next**.
Oracle Components Integration	If you selected any products to integrate with, when you configured LDAPHOST1, then select the same option here. Click **Next**.
Server Tuning	You have the option of allocating specific resources to the OUD instance. You can choose to accept the default of change the resource allocations based on your deployment. If this server is used only for OUD as in a distributed deployment then be sure to check the box Dedicated Machine for OUD. Click **Next**.
Review	Verify that the information displayed is correct. If you wish the OUD server to be started after configuration, ensure that you select the option **Start the server**.
Finished	Click **Close**.

Validating Oracle Unified Directory on LDAPHOST2

After configuration you can validate that Oracle Unified Directory is working by performing a simple search. To do this issue the following command:

```
OUD_ORACLE_INSTANCE/OUD/bin/ldapsearch -h LDAPHOST2.example.com -p 1389 -D cn=oudadmin -
b "" -s base "(objectclass=*)" supportedControl
```

If Oracle Unified Directory is working correctly, you see a list `supportedControl` entries returned.

If you have enabled SSL on the directory, you can test it using the command:

```
OUD_ORACLE_INSTANCE/OUD/bin/ldapsearch -h LDAPHOST2.example.com -p 1636 --useSSL -D
cn=oudadmin -b "" -s base "(objectclass=*)" supportedControl
```

To check that Oracle Unified Directory replication is enabled, issue the command:

```
OUD_ORACLE_INSTANCE/OUD/bin/status
```

You are prompted for the Administrator bind DN (`cn=oudadmin`) and its password.

You then see output similar to the following example. Replication is set to enable.

```
--- Server Status ---
Server Run Status: Started
Open Connections: 2

--- Server Details ---
Host Name: slc01fnv
```

```
Administrative Users: cn=oudadmin
Installation Path: /u01/oracle/products/dir/oud
Instance Path: /u02/private/oracle/config/instances/oud1/OUD
Version: Oracle Unified Directory 12.2.1.4.0
Java Version: 1.8.0_102
Administration Connector: Port 4444 (LDAPS)

--- Connection Handlers ---
Address:Port : Protocol : State
-------------:-------------:----------- :
LDIF : Disabled
8989 : Replication : Enabled
0.0.0.0:161 : SNMP : Disabled
0.0.0.0:1389 : LDAP : Enabled
0.0.0.0:1636 : LDAPS : Enabled
0.0.0.0:1689 : JMX : Disabled

--- Data Sources ---
Base DN: dc=example ,dc=com
Backend ID: userRoot
Entries: 1
Replication: Enabled
Missing Changes: 0
Age Of Oldest Missing Change: <not available>
Status
```

Installing and Configuring Oracle Unified Directory Service Manager

Oracle Unified Directory Service Manager (OUDSM) is a Graphical User Interface (GUI) tool used to manage Oracle Unified Directory.

It is not mandatory to be installed in the production environments; however, OUDSM makes managing Oracle Unified Directory easier.
It is recommended that, if you are installing OUDSM, create it in its own light-weight domain.

The following topics describe how to do it:

- Creating a Domain for Oracle Unified Directory Service Manager
  You can create a domain for Oracle Unified Directory Service Manager (OUDSM) without depending on the Oracle Database or the Repository Creation Utility (RCU) using the WebLogic Scripting Tool (WLST) command.
- Starting the Oracle Unified Directory Service Manager Domain
  After configuring the Oracle Unified Directory Service Manager domain, start the Administration Server to manage the domain.

Creating a Domain for Oracle Unified Directory Service Manager

You can create a domain for Oracle Unified Directory Service Manager (OUDSM) without depending on the Oracle Database or the Repository Creation Utility (RCU) using the WebLogic Scripting Tool (WLST) command.

> **Note:**
>
> This is the Oracle preferred approach to set up a domain for OUDSM. It is recommended not to extend this domain with any other products or components. In this approach, you do not have to run `config.sh`.

To set up the OUDSM domain using the WLST, do the following:

1. Launch the WLST by running the following command:

   On UNIX:

   ```
   DIR_ORACLE_HOME/oracle_common/common/bin/wlst.sh
   ```

2. Run the following command to create a compact domain for Oracle Unified Directory Services Manager:

   ```
   createOUDSMDomain(domainLocation=path_to_domain_home,weblogicPort=wls_port,
   weblogicSSLPort=ssl_port,weblogicUserName=wls_user,weblogicUserPassword=wls
   _password)
   ```

   In the above command, specify the values for the following parameters:

   - `domainLocation`: This is the absolute path to the domain home. For example, `PRIVATE_CONFIG_DIR/domains/OUDSMDomain`.
   - `weblogicPort`: This is the WebLogic port. This value must be unique to the server.
   - `weblogicSSLPort`: This is the WebLogic SSL port. This parameter is optional and is not enabled if not passed.
   - `weblogicUserName`: This is the WebLogic user name. This parameter is optional. If not specified, the default value `weblogic` is passed.
   - `weblogicUserPassword`: This is the WebLogic Administration Server user password.

   For example:

   ```
   createOUDSMDomain(domainLocation="/u02/private/oracle/config/domains/
   OUDSMDomain",weblogicPort=7001,weblogicSSLPort=7002,weblogicUserPassword='<pas
   sword>')
   ```

3. Enter exit() to exit out of wlst.

Starting the Oracle Unified Directory Service Manager Domain

After configuring the Oracle Unified Directory Service Manager domain, start the Administration Server to manage the domain.

To do this, complete the following steps:

1. Start the Administration Server using the following command:

   ```
   PRIVATE_CONFIG_DIR/domain/OUDSMDomain/bin/startWebLogic.sh
   ```

2. Verify that the Administration Server is up and running by accessing the Oracle Unified Directory Services Manager at the following URL:

   ```
   http://hostname:port/oudsm
   ```

   In the above command, *`hostname`* is the name of the server on which WebLogic Server is installed. *`port`* is the administrative port for the WebLogic Administration Server. The default port value is `7001`.

Configuring Oracle HTTP Server for Oracle Unified Directory Services Manager

If you want to access the Oracle Unified Directory Services Manager (OUDSM) console through Oracle Web Servers, then you must add the necessary entry to one of your administrative virtual hosts.

Once you have configured your Oracle HTTP server as described in Configuring Oracle HTTP Server for an Enterprise Deployment, then you can configure the Oracle HTTP Server to route requests to the Oracle Unified Directory Services Manager. To do this:

1. Add the following entries to the `iadadmin_vh.conf` or `igd_admin_vh.conf` files located at *WEB_DOMAIN_HOME*`/config/fmwconfig/components/OHS/ohs1/moduleconf/`:

   ```
   <Location /oudsm>
           WebLogicHost LDAPHOST1.example.com
           WebLogicPort 7001
   </Location>
   ```

   > **Note:**
   >
   > There are separate directories for configuration and runtime instance files. The runtime files under the `.../OHS/instances/ohsn/*` folder should not be edited directly. Edit only the `.../OHS/ohsn/*` configuration files.

2. Copy the `igdadmin_vh.conf` or `iadadmin_vh.conf` file to the following configuration directory of the second Oracle HTTP Server instance (ohs2):

   OHS_DOMAIN_HOME`/config/fmwconfig/components/ohs2/moduleconf/`

3. Restart the Oracle HTTP server instances on WEBHOST1 and WEBHOST2.

Preparing an Existing LDAP Directory

Before you can use an LDAP directory with Oracle Identity and Access Management, it must be extended with object classes required by Oracle Access Manager.

In addition, certain users and groups need to be seeded into the directory. These users and groups will be used by the various Oracle Identity and Access Management products as described later.

> **Note:**
>
> This procedure involves running a utility provided as part of the Oracle Identity and Access Management suite. You must have installed the software for either Oracle Access Manager (Installing the Oracle Fusion Middleware Infrastructure) or Oracle Identity Manager (Installing the Oracle Fusion Middleware Infrastructure on OIMHOST1) to continue.

This section includes the following topics:

- About the Enterprise Deployment Users and Groups
- Creating a Configuration File
- Preparing OUD as the Identity Store
- Creating Access Control Lists in Non-Oracle Directories

About the Enterprise Deployment Users and Groups

The following topics provide important information on the purpose and characteristics of the enterprise deployment administration users and groups.

- About Using Unique Administration Users for Each Domain

About Using Unique Administration Users for Each Domain

When you use a central LDAP user store, you can provision users and groups for use with multiple Oracle WebLogic Server domains. As a result, there is a possibility that one WebLogic administration user can have access to all the domains within an enterprise.

It is a best practice to create and assign a unique distinguished name (DN) within the directory tree for the users and groups that you provision for the administration of your Oracle Fusion Middleware domains.

For example, create two users called `oamLDAP` and `oimLDAP` which is used to connect the WebLogic domain to LDAP. This allows the domain to see the users and groups which exist in the directory. You can create a different user for each domain or use a single user for multiple domains. Under no circumstances should the default LDAP administration user be used for this purpose. You must create these users in the `systemids` container. This container is used for system users that are not normally visible to users. Placing the user into the `systemids` container ensures that customers who have Oracle Identity Governance do not reconcile this user.

Using a different user for Oracle Access Management (OAM) and Oracle Identity Manager (OIM) LDAP connections ensures that the user that OAM uses to connect to LDAP has a restricted privilege set.

Create a user called `weblogic_iam` and an administration group called `WLSAdministrators`. Users in the `WLSAdministrators` group will be allowed to access the following:

- Oracle Fusion Middleware Control
- Oracle WebLogic Administration Console

Create a user called `oamadmin` and an administration group called `OAMAdministrators`. Users in the `OAMAdministrators` group are allowed to access the following:

- Oracle Access Policy Manager
- Oracle Access Manager Console

Creating a Configuration File

Create a property file `iam.props`, to use when preparing the Identity Store and as a basis for later integration and configuration processes. The file will have the structure described in this section. When creating the file do not include any blank lines.

The property files in this section are complete examples. Some of the parameters specified in the file will not be used until later configuration steps in the guide. It is only necessary to include the properties for the products you are going to use.

This section includes the following topics:

- Oracle Unified Directory Example
- Explanation of Property Values

Oracle Unified Directory Example

The following is and example configuration file for Oracle Unified Directory:

```
# Common
IDSTORE_HOST: idstore.example.com
IDSTORE_PORT: 1389
IDSTORE_ADMIN_PORT: 4444
IDSTORE_KEYSTORE_FILE: OUD_INSTANCE_HOME/OUD/config/admin-keystore
IDSTORE_KEYSTORE_PASSWORD: Password key
IDSTORE_BINDDN: cn=oudadmin
IDSTORE_GROUPSEARCHBASE: cn=Groups,dc=example,dc=com
IDSTORE_SEARCHBASE: dc=example,dc=com
IDSTORE_USERNAMEATTRIBUTE: cn
IDSTORE_LOGINATTRIBUTE: uid
OAM11G_SERVER_LOGIN_ATTRIBUTE: uid
IDSTORE_USERSEARCHBASE: cn=Users,dc=example,dc=com
IDSTORE_NEW_SETUP: true
IDSTORE_DIRECTORYTYPE: OUD
# OAM
IDSTORE_OAMADMINUSER: oamadmin
IDSTORE_OAMSOFTWAREUSER: oamLDAP
OAM11G_IDSTORE_ROLE_SECURITY_ADMIN: OAMAdministrators
# OAM and OIM
IDSTORE_SYSTEMIDBASE: cn=SystemIDs,dc=example,dc=com
# OIM
IDSTORE_OIMADMINGROUP: OIMAdministrators
IDSTORE_OIMADMINUSER: oimLDAP
# WebLogic
IDSTORE_WLSADMINUSER : weblogic_iam
IDSTORE_WLSADMINGROUP : WLSAdministrators
```

Explanation of Property Values

This section explains the configuration file property values.

- LDAP Properties
- OUD Properties
- OAM Properties
- OIM Properties
- WebLogic Properties

LDAP Properties

- IDSTORE_HOST and IDSTORE_PORT are, respectively, the host and port of your Identity Store directory. When preparing the Identity Store these should point to one of the LDAP instances. When configuring components such as OAM or OIM they should point to the load balancer entry point.

- IDSTORE_DIRECTORYTYPE is the type of directory you are using. Valid value is OUD.
- IDSTORE_BINDDN is an administrative user in the Identity Store Directory
- IDSTORE_SEARCHBASE is the location in the directory where Users and Groups are stored.
- IDSTORE_LOGINATTRIBUTE is the LDAP attribute, which contains the users Login name.
- IDSTORE_USERSEARCHBASE is the location in the directory where Users are Stored.
- IDSTORE_GROUPSEARCHBASE is the location in the directory where Groups are Stored.
- IDSTORE_SYSTEMIDBASE is the location of a container in the directory where system users can be placed when you do not want them in the main user container.
- IDSTORE_USERNAMEATTRIBUTE this is the name of the LDAP attribute which stores a users name.
- IDSTORE_LOGIN_ATTRIBUTE this is the name of the LDAP attribute where userids are stored.

OUD Properties

- IDSTORE_ADMIN_PORT is the administration port of your Oracle Unified Directory instance. If you are not using Oracle Unified Directory, you can leave out this parameter.
- IDSTORE_KEYSTORE_FILE is the location of the Oracle Unified Directory Keystore file. It is used to enable communication with Oracle Unified Directory using the Oracle Unified Directory administration port. It is called `admin-keystore` and is located in *OUD_INSTANCE_HOME*`/OUD/config`. If you are not using Oracle Unified Directory, you can leave out this parameter.
- IDSTORE_KEYSTORE_PASSWORD is the encrypted password of the Oracle Unified Directory keystore. This value can be found in the file *OUD_INSTANCE_HOME*`/OUD/config/admin-keystore.pin`.
- IDSTORE_NEW_SETUP this parameter is used when preparing a directory for the first time.

OAM Properties

- IDSTORE_HOST and IDSTORE_PORT are, respectively, the host and port of your Identity Store directory. When preparing the Identity Store these should point to one of the LDAP instances. When configuring components such as OAM or OIM they should point to the load balancer entry point.
- IDSTORE_OAMADMINUSER is the name of the user you want to create as your Access Manager Administrator.
- IDSTORE_OAMSOFTWAREUSER is a user that gets created in LDAP that is used when Access Manager is running to connect to the LDAP server.
- OAM11G_IDSTORE_ROLE_SECURITY_ADMIN is the name of the group, which is used to allow access to the OAM console. Only users assigned to this group will be able to access the OAM Console.
- OAM11G_SERVER_LOGIN_ATTRIBUTE this is the name of the LDAP attribute where userids are stored, this should be the same as the IDSTORE_LOGIN_ATTRIBUTE.

> **Note:**
>
> You can create different administrator accounts and groups for each of the products or use a single administration user. The example above uses a single Administration User and Group.
>
> The OAMSOFTWAREUSER is the user that OAM uses to connect to LDAP. The OIMADMINUSER is the user that OIM uses to connect to LDAP. You can create separate users for each product or just use the same user.

OIM Properties

- IDSTORE_OIMADMINGROUP Is the name of the group you want to create to hold your Oracle Identity Governance administrative users.
- IDSTORE_OIMADMINUSER is the user that Oracle Identity Governance uses to connect to the Identity store.

> **Note:**
>
> The OAMSOFTWAREUSER is the user that OAM uses to connect to LDAP. The OIMADMINUSER is the user that OIG uses to connect to LDAP. You can create separate users for each product or just use the same user.

WebLogic Properties

- IDSTORE_WLSADMINUSER: The username to be created for logging in to the web logic domain once it is enabled by Single Sign-On.
- IDSTORE_WLSADMINGROUP: is the name of the group to which users who are allowed to log in to the WebLogic system components, such as the WLS Console and EM, belong.

Preparing OUD as the Identity Store

Before an Oracle LDAP directory can be used with Oracle Identity and Access Management, the directory needs to be pre-configured.

This process involves the creation of additional object classes within the directory and the seeding of users that the Oracle Identity and Access Management suite will use to connect to the directory. There are two phases to the configuration process:

- Pre-configure: This adds the required object classes.
- Seeding of Users.

To do this, perform the following tasks on LDAPHOST1 if you are extending Oracle Unified Directory:

- Directory Pre-Configuration
- Seeding Users and Groups
- Granting OUD changelog Access
- Updating Oracle Unified Directory ACIs

- Creating OUD Indexes

Directory Pre-Configuration

Before an Oracle LDAP directory can be used with Oracle Identity and Access Management, the directory needs to be pre-configured.

This process involves the creation of additional object classes within the directory and the seeding of users that the Oracle Identity and Access Management suite will use to connect to the directory. There are two phases to the configuration process:

- Pre-configure: This adds the required object classes.
- Seeding of Users.

To do this, perform the following tasks on LDAPHOST1 if you are extending Oracle Unified Directory:

> **Note:**
>
> The preparation of LDAP is performed using a tool called `idmConfigTool`. This tool comes bundled with the Oracle Identity and Access Management software. Before you perform the steps in this section, you must install the Oracle Identity and Access Management. See Installing the Oracle Fusion Middleware Infrastructure. The steps in this section can be run from either OAMHOST1 or OIMHOST1.
>
> If your Directory is on a different host to the *`IAD_ORACLE_HOME`*, then the `idmconfigTool.sh` tool will need to be run from that host. If you have a firewall between the *`IAD_ORACLE_HOME`* and your directory server, you will be required to open up the LDAP ports in that firewall for the duration of this step.
>
> If you are installing OIM only and wish to configure your directory, use *`IGD_ORACLE_HOME`* instead of *`IAD_ORACLE_HOME`*. The `idmtool` is the same in both the locations.

1. Set the environment variables:
   - `MW_HOME`: Set it to either *`IAD_ORACLE_HOME`* or *`IGD_ORACLE_HOME`*
   - `JAVA_HOME`: Set it to Java Home.
   - `ORACLE_HOME`: Set it to `MW_HOME/idm`
2. Configure the Identity Store using the command `idmConfigTool` from the location `ORACLE_HOME/idmtools/bin`.

   > **Note:**
   >
   > When you run the `idmConfigTool`, it creates or appends to the file `idmDomainConfig.param`. This file is generated in the same directory that the `idmConfigTool` is run from. To ensure that each time the tool is run, the same file is appended to, always run the `idmConfigTool` from the directory *`IAD_ORACLE_HOME`*`/idmtools/bin`.

The syntax of the command on Linux is:

```
idmConfigTool.sh -preConfigIDStore input_file=configfile
```

For example:

```
idmConfigTool.sh -preConfigIDStore input_file=iam.props
```

When the command runs, you are prompted to enter the password of the account you are connecting to the Identity Store with. This command might take some time to complete.

Check the log file for any errors or warnings, and correct them. The file with the name `automation.log` is created in the directory from where you run the tool.

Seeding Users and Groups

You must seed the Identity Store with users and groups that are required by the Identity Management components.

To seed the Identity Store, perform the following tasks on OAMHOST1 or OIMHOST1:

1. Set the environment variables:
   - `MW_HOME`: Set it to either *`IAD_ORACLE_HOME`* or *`IGD_ORACLE_HOME`*
   - `JAVA_HOME`: Set it to Java Home.
   - `ORACLE_HOME`: Set it to `MW_HOME/idm`

   > **Note:**
   >
   > Replace *`IAM_ORACLE_HOME`* with either *`IGD_ORACLE_HOME`* or *`IAD_ORACLE_HOME`* depending on whether the `idmConfigTool` is being run on OIMHOST1 or OAMHOST1.

2. Configure the Identity Store using the command `idmConfigTool`, at the following location:

   ```
   ORACLE_HOME/idmtools/bin
   ```

   > **Note:**
   >
   > When you run the `idmConfigTool`, it creates or appends to the file `idmDomainConfig.param`. This file is generated in the directory from which `idmConfigTool` is run. To ensure that each time you run the tool, it appends the same file, always run the `idmConfigTool` from the following directory:
   >
   > ```
   > ORACLE_HOME/idmtools/bin
   > ```

   The syntax of the command on Linux is:

   ```
   idmConfigTool.sh -prepareIDStore mode=MODE input_file=configfile
   pwd_file=passwordfile
   ```

   The value selected for `MODE` determines the type of users to be created. Possible values for `MODE` are: `OAM`, `OIM`, and `WLS`.

- In all topologies, when you enable single sign-on for your administrative consoles, you must ensure that there is a user in your Identity Store that has the permissions to log in to your WebLogic Administration Console and Oracle Enterprise Manager Fusion Middleware Control. Type:

  ```
  idmConfigTool.sh -prepareIDStore mode=WLS input_file=iam.props
  ```

- If your topology includes Access Manager, you must seed the Identity Store with users that are required by Access Manager. Type:

  ```
  idmConfigTool.sh -prepareIDStore mode=OAM input_file=iam.props
  ```

- If your topology includes Oracle Identity Governance, you must seed the Identity Store with the xelsysadm user and assign it to an Oracle Identity Governance administrative group. You must also create a user outside of the standard cn=Users location to be able to perform reconciliation. This user is also the user that should be used as the bind DN when connecting to directories with Oracle Virtual Directory. Type

  ```
  idmConfigTool.sh -prepareIDStore mode=OIM input_file=iam.props
  ```

Note:

This command also creates a container in your Identity Store for reservations.

Note:

When entering a password for `xelsysadm` ensure that it is the same at the OIM policy that is it must be at least 8 characters long, contain an Uppercase character, and a number.

When the command runs, you are prompted to enter the password of the account you are connecting to the Identity Store with.

After running each command, check the log file for any errors or warnings and correct them. The file with the name automation.log is created in the directory from where you run the tool.

Granting OUD changelog Access

If you are using Oracle Unified Directory, you must grant access to the `changelog`, by performing the following steps on `LDAPHOST1` and `LDAPHOST2`:

1. Create a file called `passwordfile` which contains the password you use to connect to OUD.
2. Remove the existing change log by issuing the command:

   ```
   OUD_ORACLE_INSTANCE/OUD/bin/dsconfig set-access-control-handler-prop \
   --remove \
   global-aci:"(target=\"ldap:///cn=changelog\")(targetattr=\"*\")(version
   3.0; acl \"External changelog access\"; deny (all) userdn=\"ldap:///
   anyone\";)"  \
   --hostname OUD Host \
   --port OUD Admin Port \
   --trustAll \
   ```

```
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

For example:

```
OUD_ORACLE_INSTANCE/OUD/bin/dsconfig set-access-control-handler-prop \
--remove \
global-aci:"(target=\"ldap:///cn=changelog\")(targetattr=\"*\")(version
3.0; acl \"External changelog access\"; deny (all) userdn=\"ldap:///
anyone\";)" \
--hostname LDAPHOST1.example.com \
--port 4444 \
--trustAll \
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

3. Add the new ACI:

```
OUD_ORACLE_INSTANCE/OUD/bin/dsconfig set-access-control-handler-prop \
--add \
global-aci:"(target=\"ldap:///cn=changelog\")(targetattr=\"*\")(version
3.0; acl \"External changelog access\"; allow
(read,search,compare,add,write,delete,export) groupdn=\"ldap:///
cn=OIMAdministrators,cn=groups,dc=example,dc=com\";)" \
--hostname OUD Host \
--port OUD Admin Port \
--trustAll \
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

For example:

```
OUD_ORACLE_INSTANCE/OUD/bin/dsconfig set-access-control-handler-prop \
--add \
global-aci:"(target=\"ldap:///cn=changelog\")(targetattr=\"*\")(version
3.0; acl \"External changelog access\"; allow
(read,search,compare,add,write,delete,export) groupdn=\"ldap:///
cn=OIMAdministrators,cn=groups,dc=example,dc=com\";)" \
--hostname LDAPHOST1.example.com \
--port 4444 \
--trustAll \
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

4. Then, add the following ACI:

```
OUD_ORACLE_INSTANCE/bin/dsconfig set-access-control-handler-prop \
--add global-aci:"(targetcontrol=\"1.3.6.1.4.1.26027.1.5.4\")(version
3.0;acl \"OIMAdministrators control access\"; allow(read)
groupdn=\"ldap:///cn=oimAdminGroup,cn=groups,dc=example,dc=com\";)" \
--hostname OUD_HOST \
--port OUD_ADMIN_PORT \
--trustAll \
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

For example:

```
OUD_ORACLE_INSTANCE/bin/dsconfig set-access-control-handler-prop \
--add global-aci:"(targetcontrol=\"1.3.6.1.4.1.26027.1.5.4\")(version
3.0;acl \"OIMAdministrators control access\"; allow(read)
groupdn=\"ldap:///cn=oimAdminGroup,cn=groups,dc=example,dc=com\";)" \
--hostname LDAPHOST1.example.com \
--port 4444 \
--trustAll \
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

5. Next, add the following ACI:

```
OUD_ORACLE_INSTANCE/bin/dsconfig set-access-control-handler-prop \
--add global-aci:"(target=\"ldap:///\")(targetscope=\"base\")
(targetattr=\"lastExternalChangelogCookie\")(version 3.0; acl \"User-
Visible lastExternalChangelog\"; allow (read,search,compare)
groupdn="ldap:///cn=OIMAdministrators,cn=groups,dc=example,dc=com\";)" \
--hostname OUD_HOST \
--port OUD_ADMIN_PORT \
--trustAll \
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

For example:

```
OUD_ORACLE_INSTANCE/bin/dsconfig set-access-control-handler-prop \
--add global-aci:"(target=\"ldap:///\")(targetscope=\"base\")
(targetattr=\"lastExternalChangelogCookie\")(version 3.0; acl \"User-
VisiblelastExternalChangelog\"; allow (read,search,compare)
groupdn=\"ldap:///cn=OIMAdministrators,cn=groups,dc=example,dc=com\";)" \
--hostname LDAPHOST1.example.com \
--port 4444 \
--trustAll \
--bindDN cn=oudadmin \
--bindPasswordFile passwordfile \
--no-prompt
```

Updating Oracle Unified Directory ACIs

The following is a workaround for an Oracle Unified Directory operations failure when OIG integration is enabled.

Update *OUD_ORACLE_INSTANCE*/OUD/config/config.ldif on all OUD instances with below changes:

> **Note:**
>
> Save a copy of the original file before editing.

1. Look for the following line:

   ```
   ds-cfg-global-aci: (targetcontrol="1.3.6.1.1.12 || 1.3.6.1.1.13.1 ||
   1.3.6.1.1.13.2 || 1.2.840.113556.1.4.319 || 1.2.826.0.1.3344810.2.3 ||
   2.16.840.1.113730.3.4.18 || 2.16.840.1.113730.3.4.9 ||
   1.2.840.113556.1.4.473 || 1.3.6.1.4.1.42.2.27.9.5.9") (version 3.0; acl
   "Authenticated users control access"; allow(read) userdn="ldap:///all";)
   ```

   Remove the Object Identifier `1.2.840.113556.1.4.319` from the above aci and add it to following aci as shown:

   ```
   ds-cfg-global-aci: (targetcontrol="2.16.840.1.113730.3.4.2 ||
   2.16.840.1.113730.3.4.17 || 2.16.840.1.113730.3.4.19 ||
   1.3.6.1.4.1.4203.1.10.2 || 1.3.6.1.4.1.42.2.27.8.5.1 ||
   2.16.840.1.113730.3.4.16 || 2.16.840.1.113894.1.8.31 ||
   1.2.840.113556.1.4.319") (version 3.0; acl "Anonymous control access";
   allow(read) userdn="ldap:///anyone";)
   ```

2. Add Object Identifiers `1.3.6.1.4.1.26027.1.5.4` and `1.3.6.1.4.1.26027.2.3.4` to the following aci as shown:

   ```
   ds-cfg-global-aci: (targetcontrol="1.3.6.1.1.12 || 1.3.6.1.1.13.1 ||
   1.3.6.1.1.13.2 || 1.2.826.0.1.3344810.2.3 || 2.16.840.1.113730.3.4.18 ||
   2.16.840.1.113730.3.4.9 || 1.2.840.113556.1.4.473 ||
   1.3.6.1.4.1.42.2.27.9.5.9 || 1.3.6.1.4.1.26027.1.5.4 ||
   1.3.6.1.4.1.26027.2.3.4") (version 3.0; acl "Authenticated users control
   access"; allow(read) userdn="ldap:///all";)
   ```

3. Restart the Oracle Unified Directory server on both LDAPHOSTs.

Creating OUD Indexes

When you ran the `idmConfigTool` to prepare an OUD identity store, it creates indexes for the data on the instance against which it is run. These indexes must be manually created on each of the OUD instances in LDAPHOST2.

To do this, run the following commands on LDAPHOST2:

```
OUD_ORACLE_INSTANCE/OUD/bin/ldapmodify -h LDAPHOST2.example.com -Z -X -p 4444
-a -D "cn=oudadmin" -j passwordfile -c \-f IAD_ORACLE_HOME/idm/oam/server/oim-
intg/ldif/ojd/schema/ojd_user_index_generic.ldif


OUD_ORACLE_INSTANCE/OUD/bin/ldapmodify -h LDAPHOST2.example.com -Z -X -p 4444
-a -D "cn=oudadmin" -j  passwordfile -c \-f IAD_ORACLE_HOME/idm/idmtools/
templates/oud/oud_indexes_extn.ldif
```

Rebuild the Indexes

Once the indexes have been created on all of the LDAP Hosts, the indexes should be rebuilt using the commands:

1. Shutdown OUD by issuing the command:

   ```
   OUD_ORACLE_INSTANCE/OUD/bin/stop-ds
   ```

2. Execute the command:

   ```
   OUD_ORACLE_INSTANCE/OUD/bin/rebuild-index --rebuildAll -b "dc=example,dc=com"
   ```

3. Restart OUD by issuing the command:

   ```
   OUD_ORACLE_INSTANCE/OUD/bin/start-ds
   ```

4. Repeat for every LDAPHOST including the host, which the idmTool was run against, to maintain availability only stop the directory for which you are rebuilding the indexes.

Creating Access Control Lists in Non-Oracle Directories

In the preceding sections, you seeded the Identity Store with users and artifacts for the Oracle components. If your Identity Store is not Oracle Unified Directory, Oracle Directory Server Enterprise Edition, you must set up the access control lists (ACLs) to provide appropriate privileges to the entities you created, this is true even if using Oracle Virtual Directory in front of them. This section lists the artifacts created and the privileges required for the artifacts.

- Systemids. The System ID container is created for storing all the system identifiers. If there is another container in which the users are to be created, that is specified as part of the admin.
- Access Manager Admin User. This user is added to the OAM Administrator group, which provides permission for the administration of the Oracle Access Management Console. No LDAP schema level privileges are required, since this is just an application user.
- Access Manager Software User. This user is added to the groups where the user gets read privileges to the container. This is also provided with schema admin privileges.
- Oracle Identity Governance user oigLDAP under System ID container. Password policies are set accordingly in the container. The passwords for the users in the System ID container must be set up so that they do not expire.
- Oracle Identity Governance administration group. The Oracle Identity Governance user is added as its member. The Oracle Identity Governance admin group is given complete read/write privileges to all the user and group entities in the directory.
- WebLogic Administrator. This is the administrator of the IDM domain for Oracle Virtual Directory

- WebLogic Administrator Group. The WebLogic administrator is added as a member. This is the administrator group of the IDM domain for Oracle Virtual Directory.
- Reserve container. Permissions are provided to the Oracle Identity Governance admin group to perform read/write operations.

13 Creating Infrastructure for Oracle Access Management

The following topics describe how to install and configure an initial domain, which can be used as the starting point for an enterprise deployment. Later chapters in this guide describe how to extend this initial domain with the various products and components that comprise the enterprise topology you are deploying.

A complete Oracle Identity and Access Management uses a split domain deployment, where there is a single domain for Oracle Access Management and a different one for Oracle Identity Governance. You must create a separate infrastructures for Access and Governance.

- About the Initial Infrastructure Domain
  Before you create the initial Infrastructure domain, ensure that you review the key concepts.
- Variables Used When Creating Infrastructure for Oracle Access Management
  As you perform the tasks in this chapter, you will be referencing the directory variables listed in this section.
- Installing the Oracle Fusion Middleware Infrastructure
  Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for Oracle Access Management.
- Installing Oracle Access Management for an Enterprise Deployment
  The procedure for installing Oracle Access Management in an enterprise deployment domain is explained in this section.
- Configuring LDAP
  It details the procedure to configure LDAP.
- Creating the Database Schemas for Access Manager
  Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain for Oracle Access Management. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.
- Configuring the Oracle Access Management Domain
  The following topics provide instructions for creating an Oracle Access Management domain using the Fusion Middleware Configuration wizard.
- Configuring the Domain Directories and Starting the Servers
  After the domain is created and the Node Manager is configured, you can then configure the additional domain directories and start the Administration Server and any Managed Servers on the AdminHost.
- Propagating the Domain and Starting the Node Manager on OAMHOST2
  After you start and validate the Administration Server and WLS_WSM1 Managed Server on OAMHOST1, you can then perform the following tasks on OAMHOST2..
- Removing OAM Server from WebLogic Server 12c defaultCoherenceCluster
  You must exclude all Oracle Access Management (OAM) clusters (including policy manager and OAM runtime server) from the default WebLogic Server 12*c* coherence cluster using the WebLogic Server Administration Console.

- Adding a Load Balancer Certificate to JDK Trust Stores
- Tuning the oamDS Data Source
  For optimium performance, increase the number of connections allowed by the OAM data source.
- Enabling Virtualization
  Use the Fusion Middleware Control to enable virtualization.
- Configuring the WebLogic Proxy Plug-In

About the Initial Infrastructure Domain

Before you create the initial Infrastructure domain, ensure that you review the key concepts.

- About the Infrastructure Distribution
- Characteristics of the Domain

About the Infrastructure Distribution

You create the initial Infrastructure domain for an enterprise deployment by using the Oracle Fusion Middleware Infrastructure distribution. This distribution contains both the Oracle WebLogic Server software and the Oracle JRF software.

The Oracle JRF software consists of Oracle Web Services Manager, Oracle Application Development Framework (Oracle ADF), Oracle Enterprise Manager Fusion Middleware Control, the Repository Creation Utility (RCU), and other libraries and technologies that are required to support the Oracle Fusion Middleware products.

> **Note:**
>
> The Access infrastructure does not use the Web Services Manager.

See Understanding Oracle Fusion Middleware Infrastructure in *Understanding Oracle Fusion Middleware*.

Characteristics of the Domain

The following table lists some of the key characteristics of the domain that you are about to create. Reviewing these characteristics helps you to understand the purpose and context of the procedures that are used to configure the domain.

Many of these characteristics are described in more detail in Understanding a Typical Enterprise Deployment.

Characteristic of the Domain	More Information
Uses a separate virtual IP (VIP) address for the Administration Server.	Configuration of the Administration Server and Managed Servers Domain Directories
Uses separate domain directories for the Administration Server and the Managed Servers in the domain.	Configuration of the Administration Server and Managed Servers Domain Directories
Uses a per domain Node Manager configuration.	About the Node Manager Configuration in a Typical Enterprise Deployment

Characteristic of the Domain	More Information
Requires a separately installed LDAP-based authentication provider.	Understanding OPSS and Requests to the Authentication and Authorization Stores

Variables Used When Creating Infrastructure for Oracle Access Management

As you perform the tasks in this chapter, you will be referencing the directory variables listed in this section.

These directory variables are defined in File System and Directory Variables Used in This Guide.

- IAD_ORACLE_HOME
- IAD_ASERVER_HOME
- IAD_MSERVER_HOME
- APPLICATION_HOME
- JAVA_HOME

In addition, you'll be referencing the following virtual IP (VIP) addresses and host names defined in Physical and Virtual IP Addresses Required by the Enterprise Topology:

- ADMINVHN
- OAMHOST1
- OAMHOST2
- DBHOST1
- DBHOST2
- SCAN Address for the Oracle RAC Database (DB-SCAN.examle.com)

> **Note:**
>
> Depending on the domain you are creating, you must add the prefix to ADMINVHN. For example, IAD_ADMINVHN.

> **Note:**
>
> The instructions in this section use the installation on OIMHOST1 and OIMHOST2 as an example. If you are creating the infrastructure domain for Access, then substitute OAMHOST1 and OAMHOST2 wherever appropriate.

Installing the Oracle Fusion Middleware Infrastructure

Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for Oracle Access Management.

- Installing a Supported JDK
- Starting the Infrastructure Installer
- Navigating the Infrastructure Installation Screens
- Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers
- Checking the Directory Structure
  After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.

Installing a Supported JDK

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system.

- Locating and Downloading the JDK Software
- Installing the JDK Software
  Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

Locating and Downloading the JDK Software

To find a certified JDK, see the certification document for your release on the Oracle Fusion Middleware Supported System Configurations page.

After you identify the Oracle JDK for the current Oracle Fusion Middleware release, you can download an Oracle JDK from the following location on Oracle Technology Network:

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the following locations:

- On the shared storage device, where it will be accessible from each of the application tier host computers, install the JDK in the location specified in File System and Directory Variables Used in This Guide.
- On the local storage device for each of the Web tier host computers. The Web tier host computers, which reside in the DMZ, do not necessarily have access to the shared storage on the application tier.
- On the local storage device for each of the directory tier host computers, in case of the directory hosts not utilizing the shared storage.

For more information about the recommended location for the JDK software, see Understanding the Recommended Directory Structure for an Enterprise Deployment.

To install JDK 1.8.0_211:

1. Change directory to the location where you downloaded the JDK archive file.

   ```
   cd download_dir
   ```

2. Unpack the archive into the JDK home directory, and then run the following commands:

   ```
   tar -xzvf jdk-8u201-linux-x64.tar.gz
   ```

   Note that the JDK version listed here was accurate at the time this document was published. For the latest supported JDK, see the *Oracle Fusion Middleware System Requirements and Specifications* for the current Oracle Fusion Middleware release.

3. Move the JDK directory to the recommended location in the directory structure.

   For example:

   ```
   mv ./jdk1.8.0_211 /u01/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

   For example:

   ```
   export JAVA_HOME=/u01/oracle/products/jdk
   export PATH=$JAVA_HOME/bin:$PATH
   ```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

   ```
   java -verison
   ```

   The Java version in the output should be displayed as “1.8.0_211”.

Starting the Infrastructure Installer

To start the installation program, perform the following steps.

1. Go to the directory where you downloaded the installation program.

2. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

   ```
   $JAVA_HOME/bin/java -d64 -jar distribution_file_name.jar
   ```

   In this example:

   - Replace *`JAVA_HOME`* with the environment variable or actual JDK location on your system.
   - Replace *`distribution_file_name`* with the actual name of the distribution JAR file.

     If you download the distribution from the Oracle Technology Network (OTN), then the JAR file is typically packaged inside a downloadable ZIP file.

     To install the software required for the initial Infrastructure domain, the distribution you want to install is:

     fmw_12.2.1.4.0_infrastructure_generic.jar.

For more information about the actual file names of each distribution, see Identifying and Obtaining Software Downloads for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation. See Navigating the Installation Screens for a description of each installation program screen.

Navigating the Infrastructure Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name or click the **Help** button on the screen.

Table 13-1 Navigating the Infrastructure Installation Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears if you are installing any Oracle product on this host for the first time. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location. See Understanding the Oracle Central Inventory in *Installing Software with the Oracle Universal Installer*. **Note:** Oracle recommends that you configure the central inventory directory on the products shared volume. Example: `/u01/oracle/products/oraInventory` You may also need to execute the `createCentralinventory.sh` script as root from the `oraInventory` folder after the installer completes.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to search My Oracle Support automatically for available patches or automatically search a local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For the purposes of an enterprise deployment, enter the value of the *IGD_ORACLE_HOME* variable listed in Table 8-2.
Installation Type	Use this screen to select the type of installation and as a consequence, the products and feature sets that you want to install. For this topology, select **Fusion Middleware Infrastructure**. **Note:** The topology in this document does not include server examples. Oracle strongly recommends that you do not install the examples into a production environment.

Table 13-1 (Cont.) Navigating the Infrastructure Installation Screens

Screen	Description
Prerequisite Checks	This screen verifies that your system meets the minimum requirements. If there are any warning or error messages, refer to the Oracle Fusion Middleware System Requirements and Specifications document on the Oracle Technology Network (OTN).
Security Updates	If you already have an Oracle Support account, use this screen to indicate how you would like to receive security updates. If you do not have one and are sure that you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.
Installation Summary	Use this screen to verify the installation options that you have selected. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file. Response files can be used later in a silent installation situation. For more information about silent or command-line installation, see Using the Oracle Universal Installer in Silent Mode in *Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click **Finish** to dismiss the installer.

Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers

If you have configured a separate shared storage volume or partition for secondary hosts, then you must install the Infrastructure on one of those hosts.

See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

To install the software on the other host computers in the topology, log in to each host, and use the instructions in Starting the Infrastructure Installer and Navigating the Infrastructure Installation Screens to create the Oracle home on the appropriate storage device.

Checking the Directory Structure

After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.

To check the directory structure:

1. Change to the *ORACLE_HOME* directory where you installed the Infrastructure.
2. Enter the following command:

   ```
   ls --format=single-column
   ```

   The directory structure on your system must match the structure shown in the following example:

```
cfgtoollogs
coherence
em
inventory
OPatch
oracle_common
oraInst.loc
oui
wlserver
```

See What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

Installing Oracle Access Management for an Enterprise Deployment

The procedure for installing Oracle Access Management in an enterprise deployment domain is explained in this section.

This section contains the following procedures.

- Starting the Oracle Identity and Access Management Installation Program
- Navigating the Installation Screens
- Installing Oracle Access Management on the Other Host Computers
- Verifying the Installation

Starting the Oracle Identity and Access Management Installation Program

To start the installation program:

1. Log in to OAMHOST1.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

   ```
   JAVA_HOME/bin/java -d64 -jar fmw_12.2.1.4.0_idm_generic.jar
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

When the installation program appears, you are ready to begin the installation.

Navigating the Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name.

Screen	Description
Installation Inventory Screen	If you did not create a central inventory when you installed the Oracle Fusion Middleware Infrastructure software, then this dialog box appears. Edit the **Inventory Directory** field so it points to the location of your local inventory, and then click **OK**.

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you've already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For Oracle Identity and Access Management, this should be set to *IAD_ORACLE_HOME*. For more information about Oracle Fusion Middleware directory structure, see "Selecting Directories for Installation and Configuration" in *Planning an Installation of Oracle Fusion Middleware*.
Installation Type	Use this screen to choose the type of installation you wish to deploy. You have two options: • Standalone Oracle Identity and Access Manager (Managed independently of Weblogic Server): Use this option if you are going to run Oracle Identity and Access Management with a webserver other than Weblogic. • Collocated Oracle Identity and Access Manager (Managed through WebLogic Server): Use this option if you have installed Oracle Weblogic Server into IAD_ORACLE_HOME_HOME as part of the infrastructure deployment. For Oracle Enterprise deployments, it is recommended that this option be selected.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the documents in the Roadmap for Verifying Your System Environment section in *Planning Your Oracle Fusion Middleware Infrastructure Installation.*
Installation Summary	Use this screen to verify the installation options you selected. Click **Install** to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click **Next** when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click **Finish** to dismiss the installer.

Installing Oracle Access Management on the Other Host Computers

If you have followed the EDG shared storage recommendations, there is a separate shared storage volume for product installations on IAMHOST2, and you must also install the software on IAMHOST2. See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- Reviewing the Installation Log Files
- Checking the Directory Structure
- Viewing the Contents of Your Oracle Home

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see Understanding Installation Log Files in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

The contents of your installation vary based on the options you selected during the installation.

The addition of Oracle Identity and Access Management will add the following directory and sub-directories:

```
IAD_ORACLE_HOME/
OPatch
cfgtoollogs
coherence
em
idm
inventory
oraInst.loc
oracle_common
oui
wlserver

idm/
clone
common
connectors
designconsole
idmdiag
idmtools
jlib
libovd
mbeans
modules
oam
oic
opam-connectors
plugins
remote_manager
schema
server
upgrade
```

For more information about the directory structure you should see after installation, see "What are the Key Oracle Fusion Middleware Directories?" in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See Viewing the contents of an Oracle home in *Installing Software with the Oracle Universal Installer*.

Configuring LDAP

It details the procedure to configure LDAP.

If you haven't already done so, you now need to configure your LDAP directory. To do this follow the steps in Preparing an Existing LDAP Directory.

Creating the Database Schemas for Access Manager

Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain for Oracle Access Management. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.

- Metadata Services (MDS)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Oracle Platform Security Services (OPSS)
- User Messaging Service (UMS)
- WebLogic Services (WLS)
- Common Infrastructure Services (STB)
- Oracle Access Manager (OAM)

Use the Repository Creation Utility (RCU) to create the schemas. This utility is installed in the Oracle home for each Oracle Fusion Middleware product. For more information about RCU and how the schemas are created and stored in the database, see Preparing for Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Complete the following steps to install the required schemas:

- Installing and Configuring a Certified Database
- Starting the Repository Creation Utility (RCU)
- Navigating the RCU Screens to Create the Schemas
- Verifying Schema Access

Installing and Configuring a Certified Database

Make sure that you have installed and configured a certified database, and that the database is up and running.

See the Preparing the Database for an Enterprise Deployment.

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Set the `JAVA_HOME` environment variable so it references the location where you installed a supported JDK.

See File System and Directory Variables Used in This Guide.

2. Navigate to the following directory on OAMHOSt1:

   ```
   ORACLE_HOME/oracle_common/bin
   ```

3. Start RCU:

   ```
   ./rcu
   ```

> **Note:**
>
> If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces created by the RCU, provide the `-encryptTablespace true` option when you start the RCU.
>
> This will default the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See Encrypting Tablespaces in *Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

Follow the instructions in this section to create the schemas for the Fusion Middleware Infrastructure domain:

Task 1 Introducing RCU
Review the Welcome screen and verify the version number for RCU. Click **Next** to begin.

Task 2 Selecting a Method of Schema Creation
If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load** on the Create Repository screen. The procedure in this document assumes that you have the necessary privileges.
If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option will generate a SQL script, which can be provided to your database administrator. See Understanding System Load and Product Load in *Creating Schemas with the Repository Creation Utility*.
Click **Next**.

> **Tip:**
>
> For more information about the options on this screen, see Create repository in *Creating Schemas with the Repository Creation Utility*.

Task 3 Providing Database Connection Details
Provide the database connection details for RCU to connect to your database.

1. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
2. Enter the **Port** number of the RAC database scan listener, for example 1521.
3. Enter the RAC **Service Name** of the database.

4. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.
5. Enter the **Password** of the user name that you provided in step 4.
6. If you have selected the SYS user, ensure that you set the role to SYSDBA.
7. Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

> **Tip:**
>
> For more information about the options on this screen, see Database Connection Details in *Creating Schemas with the Repository Creation Utility*.

Task 4 Specifying a Custom Prefix and Selecting Schemas

1. Specify the custom prefix you want to use to identify the Oracle Fusion Middleware schemas.

   The custom prefix is used to logically group these schemas together for use in this domain. For Oracle Access Management, use the prefix `IAD`.

   > **Tip:**
   >
   > Make a note of the custom prefix you choose to enter here; you will need this later, during the domain creation process.
   >
   > For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.

2. Select the following schemas from the list of components:
   - **AS Common Schemas**

     When you select **AS Common Schemas**, all of the schemas in this section are automatically selected. If the schemas in this section are not automatically selected, then select the required schemas.
     - Metadata Services (MDS)
     - Audit Services (IAU)
     - Audit Services Append (IAU_APPEND)
     - Audit Services Viewer (IAU_VIEWER)
     - Oracle Platform Security Services (OPSS)
     - User Messaging Service (UMS)
     - WebLogic Services (WLS)
     - Common Infrastructure Services (STB)
   - Expand the group **IDM Schemas**, and then select the **Oracle Access Manager** schema.

There are two mandatory schemas that are selected by default. You cannot deselect them: **Common Infrastructure Services** (the STB schema) and **WebLogic Services** (the WLS schema). The **Common Infrastructure Services** schema enables you to retrieve information from RCU during domain configuration. See Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.

> **Tip:**
>
> For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Click **Next** to proceed, then click **OK** on the dialog window confirming that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords
Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database security requirements before you continue. RCU will proceed at this point even if you do not meet the password polices. Hence, perform this check outside RCU itself.
Click **Next**.

> **Tip:**
>
> You must make a note of the passwords you set on this screen; you will need them later on during the domain creation process.

Task 6 Verifying the Tablespaces for the Required Schemas
You can accept the default settings on the remaining screens, or you can customize how RCU creates and uses the required tablespaces for the Oracle Fusion Middleware schemas.

> **Note:**
>
> You can configure a Fusion Middleware component to use JDBC stores for JMS servers and Transaction Logs, by using the Configuration Wizard. These JDBC stores are placed in the Weblogic Services component tablespace. If your environment expects to have a high level of transactions and/or JMS activity, you can increase the default size of the <*PREFIX*>_WLS tablespace to better suit the environment load.

Click **Next** to continue, and then click **OK** on the dialog window to confirm the tablespace creation.
For more information about RCU and its features and concepts, see About the Repository Creation Utility in *Creating Schemas with the Repository Creation Utility*.

Task 7 Creating Schemas
Review the summary of the schemas to be loaded and click **Create** to complete schema creation.

> **Note:**
>
> If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and re-create the schemas before you continue.

Task 8 Reviewing Completion Summary and Completing RCU Execution
When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users are created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

```
sqlplus <RCU_PREFIX>_OAM/<PASSWORD>@//<SCAN_ADDRESS>:<PORT>/<SERVICE_NAME>
```

For example:

```
sqlplus IADEDG_OAM/<password>@//db-scan.example.com:1521/oampdb_s.example.com
```

The output appears as follows:

```
SQL*Plus: Release 18.0.0.0.0 - Production on Mon Aug 9 01:53:57 2021
Version 18.5.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Last Successful login time: Mon Aug 09 2021 01:52:44 -07:00

Connected to:
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production
Version 18.5.0.0.0

SQL>
```

Configuring the Oracle Access Management Domain

The following topics provide instructions for creating an Oracle Access Management domain using the Fusion Middleware Configuration wizard.

For more information on other methods available for domain creation, see Additional Tools for Creating, Extending, and Managing WebLogic Domains in *Creating WebLogic Domains Using the Configuration Wizard*.

- Starting the Configuration Wizard
- Navigating the Configuration Wizard Screens to Configure Oracle Access Management Domain

Starting the Configuration Wizard

To begin domain configuration, run the following command in the Oracle Fusion Middleware Oracle home.

```
IAD_ORACLE_HOME/oracle_common/common/bin/config.sh
```

Navigating the Configuration Wizard Screens to Configure Oracle Access Management Domain

Follow the instructions in the following sections to create and configure the domain for the topology with static clusters.

> **Note:**
>
> Oracle Access Management does not support Dynamic Clusters.

- Creating the Domain with Static Clusters

Creating the Domain with Static Clusters

Follow the instructions in this section to create and configure the Oracle Access Management domain for the topology.

Domain creation and configuration includes the following tasks.

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a new domain**.
In the **Domain Location** field, specify the value of the *IAD_ASERVER_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen of the Configuration Wizard, see Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 2 Selecting the Configuration Templates
On the Templates screen, make sure **Create Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Access Management Suite - 12.2.1.4.0[idm]**
- Selecting this template automatically selects the following dependencies:
  - **Oracle Enterprise Manager - 12.2.1.4.0[em]**

- **Oracle JRF - 12.2.1.4.0[oracle_common]**
- **WebLogic Coherence Cluster Extension - 12.2.1.4.0[wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 3 Selecting the Application Home Location
On the Application Location screen, specify the value of the *APPLICATION_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the options on this screen can be found in Application Location in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 4 Configuring the Administrator Account
On the Administrator Account screen, specify the user name and password for the default WebLogic Administrator account for the domain.
Make a note of the user name and password specified on this screen; you will need these credentials later to boot and connect to the domain's Administration Server.
Click **Next**.

Task 5 Specifying the Domain Mode and JDK
On the Domain Mode and JDK screen:

- Select **Production** in the Domain Mode field.
- Select the **Oracle Hotspot** JDK in the JDK field.

Selecting **Production Mode** on this screen gives your environment a higher degree of security, requiring a user name and password to deploy applications and to start the Administration Server.

> **Tip:**
>
> More information about the options on this screen, including the differences between development mode and production mode, can be found in Domain Mode and JDK in *Creating WebLogic Domains Using the Configuration Wizard*.
> In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. See Creating the boot.properties File.

Click **Next**.

Task 6 Specifying the Database Configuration Type
On the Database Configuration Type screen:

- Select **RCU Data** to activate the fields on this screen.

  The **RCU Data** option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for the schemas needed to configure the domain.
- Verify that **Vendor** is `Oracle` and **Driver** is `*Oracle's Driver (Thin) for Service Connections; Versions: Any`.
- Verify that **Connection Parameters** is selected.

> **Note:**
>
> If you choose to select **Manual Configuration** on this screen, you will have to manually fill in the parameters for your schema on the JDBC Component Schema screen.

After you select **RCU Data**, fill in the fields as shown in the following table:

Field	Description
Host Name	Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the *Enterprise Deployment Workbook*. For information about the Enterprise Deployment Workbook, see Using the Enterprise Deployment Workbook.
DBMS/Service	Enter the service name for the Oracle RAC database appropriate for this domain where you will install the product schemas. For example: `iamedg.example.com` Specify the service name based on the value configured earlier in the Preparing the Database for an Enterprise Deployment section.
Port	Enter the port number on which the database listens. For example, `1521`.
Schema Owner	Enter the user name and password for connecting to the database's Service Table schema.
Schema Password	This is the schema user name and password that was specified for the Service Table component on the "Schema Passwords" screen in RCU (see Creating the Database Schemas). The default user name is *`prefix`*`_STB`, where *`prefix`* is the custom prefix that you defined in RCU.

Click **Get RCU Configuration** when you are finished specifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Click **Next** if the connection to the database is successful.

> **Tip:**
>
> More information about the **RCU Data** option can be found in Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*. More information about the other options on this screen can be found in Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 7 Specifying JDBC Component Schema Information

Verify that the values on the JDBC Component Schema screen are correct for all schemas. The schema table should be populated, because you selected **Get RCU Data** on the previous screen. As a result, the Configuration Wizard locates the database connection values for all the schemas required for this domain.

At this point, the values are configured to connect to a single-instance database. However, for an enterprise deployment, you should use a highly available Real Application Clusters (RAC) database, as described in Preparing the Database for an Enterprise Deployment.

In addition, Oracle recommends that you use an Active GridLink datasource for each of the component schemas. For more information about the advantages of using GridLink data sources to connect to a RAC database, see Database Considerations in the *High Availability Guide*.

To convert the data sources to GridLink:

1. Select all the schemas by selecting the checkbox at in the first header row of the schema table.
2. Click **Convert to GridLink** and click **Next**.

Task 8 Providing the GridLink Oracle RAC Database Connection Details

On the GridLink Oracle RAC Component Schema screen, provide the information required to connect to the RAC database and component schemas, as shown in following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, `1521`)
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the **Port** field, enter the ONS Remote port (typically, `6200`).
Enable Fan	Verify that the **Enable Fan** check box is selected, so the database can receive and process FAN events.

For more information about specifying the information on this screen, as well as information about how to identify the correct SCAN address, see Configuring Active GridLink Data Sources with Oracle RAC in the *High Availability Guide*.

You can also click **Help** to display a brief description of each field on the screen.

Click **Next**.

Task 9 Testing the JDBC Connections

Use the JDBC Component Schema Test screen to test the data source connections you have just configured.

A green check mark in the Status column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> More information about the other options on this screen can be found in Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*

Click **Next**.

Task 10 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Administration Server**

  This is required to properly configure the listen address of the Administration Server.
- **Node Manager**

  This is required to configure Node Manager.
- **Topology**

  This is required to add, delete, or modify the Settings for Server Templates, Managed Servers, Clusters, Virtual Targets, and Coherence.

> **Note:**
>
> When using the Advanced Configuration screen in the Configuration Wizard:
>
> - If any of the above options are not available on the screen, then return to the Templates screen, and be sure you selected the required templates for this topology.
> - Do not select the **Domain Frontend Host Capture** advanced configuration option. You will later configure the frontend host property for specific clusters, rather than for the domain.

Click **Next**.

Task 11 Configuring the Administration Server Listen Address
On the Administration Server screen:

1. In the **Server Name** field, retain the default value - AdminServer.
2. In the **Listen Address** field, enter the virtual host name that corresponds to the VIP of the ADMINVHN that you procured in Procuring Resources for an Enterprise Deployment and enabled in Preparing the Host Computers for an Enterprise Deployment.

   For more information on the reasons for using the ADMINVHN virtual host, see Reserving the Required IP Addresses for an Enterprise Deployment.
3. In the **Listen Port** field, enter the port number to access the administration server. This guide recommends you to use the default port `7001` for Access.

Leave the other fields at their default values. In particular, be sure that no server groups are assigned to the Administration Server.

Click **Next**.

Task 12 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type, then specify the following Node Manager credentials you will use to connect to the Node Manager:

- Username: This is the user name used to connect to the Node Manager. For example, `admin`.
- Password and Confirm Password: Enter the password you wish to associate with the Node Manager username.

> **Tip:**
>
> For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.
> For more information about per domain and per host Node Manager implementations, see About the Node Manager Configuration in a Typical Enterprise Deployment.
> For information about Node Manager configurations, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Click **Next**.

Task 13 Configuring Managed Servers
On the Managed Servers screen, a new Managed Server for Oracle Access Management appears in the list of servers.
Perform the following tasks to modify the default Oracle Access Management Managed Server and create a second Managed Server:

1. Rename the default Managed Server `oam_server1` to `WLS_OAM1`.
2. Rename the default Managed Server `oam_policy_mngr1` to `WLS_AMA1`.
3. Click **Add** to create a new Managed Server and name it `WLS_OAM2`.

   > **Tip:**
   >
   > The server names recommended here will be used throughout this document; if you choose different names, be sure to replace them as needed.

4. Use the information in the following table to fill in the rest of the columns for each Oracle Access Manager Server.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_OAM1	OAMHOST1	14100	Unchecked	Disabled	OAM-MGD-SVRS
WLS_OAM2	OAMHOST2	14100	Unchecked	Disabled	OAM-MGD-SVRS
WLS_AMA1	OAMHOST1	14150	Unchecked	Disabled	OAM-POLICY-MANAGED-SERVER
WLS_AMA2	OAMHOST2	14150	Unchecked	Disabled	OAM-POLICY-MANAGED-SERVER

> **Tip:**
>
> More information about the options on the Managed Server screen can be found in Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 14 Configuring a Cluster

In this task, you create clusters of Managed Servers to which you can target the Oracle Access Manager software.

You must create the following clusters:

Cluster	Frontend Host	Frontend HTTP Port	Frontend HTTPS Port
OAM_Cluster	login.example.com		443
AMA_Cluster	iadadmin.example.com	80	

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `OAM_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `Unspecified`.
4. Specify `login.example.com` for the **Frontend Host** field.
5. Specify `443` for the **Frontend HTTPS Port** field.

   > **Note:**
   >
   > By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Administering Clusters for Oracle WebLogic Server*.

   > **Tip:**
   >
   > More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

6. Repeat the steps to create the second cluster `AMA_Cluster`.
7. Click **Next**.

> **Tips:**
>
> For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 15 Assigning Server Templates
Click **Next** to proceed to the next screen.

Task 16 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 17 Assigning Managed Servers to the Cluster
Use the Assign Servers to Clusters screen to assign your managed servers to the clusters you have just created. At the end of this you will have the following assignments:

Cluster	Managed Servers
OAM_Cluster	WLS_OAM1 WLS_OAM2
AMA_Cluster	WLS_AMA1 WLS_AMA2

1. In the Clusters pane, select the cluster to which you want to assign the servers.
2. In the Servers pane, assign the managed servers to the clusters as in the table above, using one of the following methods:
   - Click once on the Managed Server to select it, then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click on managed server to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign each managed server to a cluster as shown in the table.
4. Click **Next** to proceed to the next screen.

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain.
In the **Cluster Listen Port**, enter `9991`.

> **Note:**
>
> For Coherence licensing information, Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Click **Next**.

Task 19 Creating Machines for Oracle Access Management Servers
Use the Machines screen to create new machines in the domain. A machine is required in order for the Node Manager to be able to start and stop the servers.
You must create a machine even if your topology contains just the Administration Server. To do this:

1. On the **Unix Machines** tab, click the **Add** button.
2. Enter OAMHOST1 in the **Name** field.
3. Enter the host name of OAMHOST1 for the Node Manage Listener address. Leave the Node Manager port to the default value of `5556`.
4. Repeat the above steps for OAMHOST2.

Under the **Unix Machine** tab, verify the names of the machines you created when creating the initial Infrastructure domain, as shown in the following table.
Click **Next** to proceed.

Name	Node Manager Listen Address	Node Manager Listen Port
ADMINHOST	Enter the value of the ADMINVHN variable.	5556
OAMHOST1	The value of the OAMHOST1 host name variable. For example, `OAMHOST1.example.com`.	5556
OAMHOST2	The value of the OAMHOST2 host name variable. For example, `OAMHOST2.example.com`.	5556

> **Tip:**
>
> More information about the options on this screen can be found in Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 20 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Oracle Access Manager Managed Servers you just created to the corresponding machines in the domain. You can assign the machines as follows:

Servers	Machines
AdminServer	ADMINHOST
WLS_AMA1 WLS_OAM1	OAMHOST1
WLS_AMA2 WLS_OAM2	OAMHOST2

> **Tip:**
>
> More information about the options on this screen can be found in Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 21 Creating Virtual Targets
Click **Next**.

Task 22 Creating Partitions
Click **Next**.

Task 23 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Create**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 24 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Configuring the Domain Directories and Starting the Servers

After the domain is created and the Node Manager is configured, you can then configure the additional domain directories and start the Administration Server and any Managed Servers on the AdminHost.

- Starting the Node Manager in the Administration Server Domain Home
  Use these steps to start the per-domain Node Manager for the *IAD_ASERVER_HOME* domain directory.
- Creating the boot.properties File
  You must create a `boot.properties` if you want to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.
- Performing the Post-Configuration Tasks for Oracle Access Management Domain
  Complete the post-configuration tasks for Oracle Access Management domain.
- Starting the Administration Server Using the Node Manager
  After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.

- Validating the Administration Server
  Before proceeding with the configuration steps, validate that the Administration Server has started successfully by making sure you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control, which both are installed and configured on the Administration Servers.
- Creating a Separate Domain Directory for Managed Servers
  When you initially create the domain for enterprise deployment, the domain directory resides on a shared disk. This default domain directory will be used to run the Administration Server. You can now create a copy of the domain on the local storage for each of your managed server hosts. The domain directory on the local (or private) storage will be used to run the Managed Servers.
- Starting the Node Manager in the Managed Server Domain Directory on OAMHOST1

Starting the Node Manager in the Administration Server Domain Home

Use these steps to start the per-domain Node Manager for the *IAD_ASERVER_HOME* domain directory.

1. Verify that the listen address in the `nodemanager.properties` file is set correctly.
   a. Open the nodemanager.properties file for editing:

      ```
      vi IAD_ASERVER_HOME/nodemanager/nodemanager.properties
      ```

   b. Make sure the `ListenAddress` property is set to the value of the ADMINVHN virtual IP address.
   c. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change to the following directory:

   ```
   cd IAD_ASERVER_HOME/bin
   ```

3. Start the Node Manager by entering the following command:

   ```
   nohup ./startNodeManager.sh > IAD_ASERVER_HOME/nodemanager/nodemanager.out 2>&1 &
   ```

   For more information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Creating the boot.properties File

You must create a `boot.properties` if you want to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.

To create a `boot.properties` file for the Administration Server:

1. Create the following directory structure:

   ```
   mkdir -p IAD_ASERVER_HOME/servers/AdminServer/security
   ```

2. In a text editor, create a file called `boot.properties` in the `security` directory that you created in the previous step, and enter the Administration Server credentials that you defined when you ran the Configuration Wizard to create the domain:

   ```
   username=adminuser
   password=password
   ```

   > **Note:**
   >
   > When you start the Administration Server, the `username` and `password` entries in the file are encrypted.
   >
   > For security reasons, minimize the amount of time the entries in the file are left unencrypted; after you edit the file, you should start the server as soon as possible so that the entries are encrypted.

3. Save the file and close the editor.

Performing the Post-Configuration Tasks for Oracle Access Management Domain

Complete the post-configuration tasks for Oracle Access Management domain.

Topics:

- Disabling the Derby Database
- Enabling the Managed Servers to use IPv6 Networking
  If the Managed Server is configured to use IPv6 networking, then you may encounter issues when you start the Managed Server.
- Setting the Memory Parameters in IAMAccessDomain
  The initial startup parameter in the IAMAccessDomain, which defines the memory usage, is insufficient. You must increase the value of this parameter.

Disabling the Derby Database

Disable the embedded Derby database, which is a file-based database, packaged with Oracle WebLogic Server. The Derby database is used primarily for development environments. As a result, you must disable it when you are configuring a production-ready enterprise deployment environment; otherwise, the Derby database process starts automatically when you start the Managed Servers.

To disable the Derby database:

1. Navigate to the following directory in the Oracle home:

   ```
   cd WL_HOME/common/derby/lib
   ```

2. Rename the Derby library jar file:

   ```
   mv derby.jar disable_derby.jar
   ```

3. If each host uses a separate file system, repeat steps 1 and 2 on each host.

Enabling the Managed Servers to use IPv6 Networking

If the Managed Server is configured to use IPv6 networking, then you may encounter issues when you start the Managed Server.

To do this, complete the following steps:

1. Edit the *IAD_ASERVER_HOME*/bin/setUserOverrides.sh file to add the following line:

   ```
   JAVA_OPTIONS="${JAVA_OPTIONS} -Djava.net.preferIPv4Stack=true"
   ```

   > **Note:**
   >
   > If the file does not exist, then create it.

2. Save and close the file.

Setting the Memory Parameters in IAMAccessDomain

The initial startup parameter in the IAMAccessDomain, which defines the memory usage, is insufficient. You must increase the value of this parameter.

To change the memory allocation setting, do the following:

1. Change the following memory allocation in the *IAD_ASERVER_HOME*/bin/setUserOverrides.sh file, by updating the Java maximum memory allocation pool (Xmx) to 8192m and initial memory allocation pool (Xms) to 1024m. For example, add the following line to be:

   ```
   MEM_ARGS="-Xms4096m -Xmx8192m"
   ```

2. Save and close the file.

Starting the Administration Server Using the Node Manager

After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.

To start the Administration Server by using the Node Manager:

1. Start the WebLogic Scripting Tool (WLST):

   ```
   cd ORACLE_COMMON_HOME/common/bin
   ./wlst.sh
   ```

2. Connect to Node Manager by using the Node Manager credentials:

   ```
   wls:/offline>nmConnect('nodemanager_username','nodemanager_password',
                'ADMINVHN','5556','domain_name',
                'IAD_ASERVER_HOME')
   ```

> **Note:**
>
> This user name and password are used only to authenticate connections between Node Manager and clients. They are independent of the server administrator ID and password and are stored in the `nm_password.properties` file located in the following directory:
>
> *`IAD_ASERVER_HOME`*`/config/nodemanager`

3. Start the Administration Server:

   ```
   nmStart('AdminServer')
   ```

> **Note:**
>
> When you start the Administration Server, it attempts to connect to Oracle Web Services Manager for WebServices policies. It is expected that the WSM-PM Managed Servers are not yet started, and so, the following message appears in the Administration Server log:
>
> ```
> <Warning><oracle.wsm.resources.policymanager>
> <WSM-02141><Unable to connect to the policy access service due to
> Oracle WSM policy manager host server being down.>
> ```

4. Exit WLST:

   ```
   exit()
   ```

Validating the Administration Server

Before proceeding with the configuration steps, validate that the Administration Server has started successfully by making sure you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control, which both are installed and configured on the Administration Servers.

To navigate to Fusion Middleware Control, enter the following URL, and log in with the Oracle WebLogic Server administrator credentials:

```
http://adminvhn:7001/em
```

To navigate to the Oracle WebLogic Server Administration Console, enter the following URL, and log in with the same administration credentials:

```
http://adminvhn:7001/console
```

Creating a Separate Domain Directory for Managed Servers

When you initially create the domain for enterprise deployment, the domain directory resides on a shared disk. This default domain directory will be used to run the Administration Server. You can now create a copy of the domain on the local storage for each of your managed server hosts. The domain directory on the local (or private) storage will be used to run the Managed Servers.

> **Note:**
>
> If you are creating a domain for Oracle Access Management, it is not necessary to perform this step at this time. This is because, at the time of infrastructure creation, there are no managed servers in existence yet.

Placing the *IAD_MSERVER_HOME* on local storage is recommended to eliminate the potential contention and overhead cause by servers writing logs to shared storage. It is also faster to load classes and jars need from the domain directory, so any temporary or cache data that Managed Servers use from the domain directory is processed quicker.

As described in Preparing the File System for an Enterprise Deployment, the path to the Administration Server domain home is represented by the *IAD_ASERVER_HOME* variable, and the path to the Managed Server domain home is represented by the *IAD_MSERVER_HOME* variable.

To create the Managed Server domain directory:

1. Sign in to the host running the Administration Server, for example, OAMHOST1, and run the `pack` command to create a template as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./pack.sh -managed=true \
             -domain=IAD_ASERVER_HOME \
             -template=/full_path/edgdomaintemplate.jar \
             -template_name=edg_domain_template \
         -log_priority=DEBUG \
             -log=/tmp/pack.log
   ```

   In this example:

   - Replace *IAD_ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.
   - Replace *full_path* with the complete path to the location where you want to create the domain template jar file. You will need to reference this location when you copy or unpack the domain template jar file. It is recommended to choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

     You must specify a full path for the template jar file as part of the `-template` argument to the pack command:

     ```
     SHARED_CONFIG_DIR/domains/template_filename.jar
     ```

   - `edgdomaintemplate.jar` is a sample name for the jar file you are creating, which will contain the domain configuration files.
   - `edg_domain_template` is the label assigned to the template data stored in the template file.

2. Make a note of the location of the `edgdomaintemplate.jar` file you just created with the pack command.

> **Tip:**
>
> For more information about the pack and unpack commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

3. If you haven't already, create the recommended directory structure for the Managed Server domain on the OAMHOST1 local storage device.

   Use the examples in File System and Directory Variables Used in This Guide as a guide.

4. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=IAD_MSERVER_HOME \
               -overwrite_domain=true \
               -template=/full_path/edgdomaintemplate.jar \
           -log_priority=DEBUG \
               -log=/tmp/unpack.log \
               -app_dir=APPLICATION_HOME
   ```

   > **Note:**
   >
   > The `-overwrite_domain` option in the unpack command allows unpacking a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this unpack operation.
   >
   > Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional JAVA command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the pack and unpack commands.

   In this example:

   - Replace *IAD_MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
   - Replace `/full_path/edgdomaintemplate.jar` with the complete path and file name of the domain template jar file that you created when you ran the pack command to pack up the domain on the shared storage device.
   - Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the pack and unpack commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

5. Change directory to the newly created Managed Server directory and verify that the domain configuration files were copied to the correct location on the OAMHOST1 local storage device.

Starting the Node Manager in the Managed Server Domain Directory on OAMHOST1

After you create the Managed Server domain directory, there are two domain home directories and two corresponding Node Manager instances on OAMHOST1. You use one Node Manager to control the Administration Server, running from Administration Server domain home, and you use the other Node Manager to control the Managed Servers, running from the Managed Server domain home.

You must start the two Node Managers independently.

> **Note:**
>
> The Node Manager for the Managed Server's *IAD_MSERVER_HOME* will be reset every time the domain configuration is unpacked. The `ListenAddress` will be changed to the *ADMINVHN* instead of the correct hostname. This needs to be changed to the correct value before starting the Node Manager service after an unpack is performed.

Follow these steps to update and start the Node Manager from the Managed Server home:

1. Verify that the listen address in the nodemanager.properties file is set correctly, by completing the following steps:
   a. Change to the following directory:

      `IAD_MSERVER_HOME/nodemanager/`

   b. Open the nodemanager.properties file for editing.
   c. Update the `ListenAddress` property to the correct hostname as follows:

      ```
      OAMHOST1: ListenAddress=OAMHOST1
      ```

   d. Update the `ListenPort` property with the correct Listen Port details.
   e. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change to the following directory:

   `IAD_MSERVER_HOME/bin`

3. Use the following command to start the Node Manager:

```
nohup ./startNodeManager.sh > IAD_MSERVER_HOME/nodemanager/nodemanager.out 2>&1 &
```

For information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Propagating the Domain and Starting the Node Manager on OAMHOST2

After you start and validate the Administration Server and WLS_WSM1 Managed Server on OAMHOST1, you can then perform the following tasks on OAMHOST2..

- Unpacking the Domain Configuration on OAMHOST2
- Starting the Node Manager in the Managed Server Domain Directory on OAMHOST2

Unpacking the Domain Configuration on OAMHOST2

Now that you have the Administration Server and the first WLS_WSM1 Managed Server running on OAMHOST1, you can configure the domain on OAMHOST2.

1. Log in to OAMHOST2.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the OAMHOST2 storage device.

   Use the examples in File System and Directory Variables Used in This Guide as a guide.
3. Make sure the `oimdomaintemplate.jar` accessible to OAMHOST2.

   For example, if you are using a separate shared storage volume or partition for OAMHOST2, then copy the template to the volume or partition mounted to OAMHOST2.
4. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=IAD_MSERVER_HOME
               -overwrite_domain=true
               -template=/full_path/create_domain.jar
               -log_priority=DEBUG
               -log=/tmp/unpack.log
               -app_dir=APPLICATION_HOME
   ```

   In this example:

   - Replace *IAD_MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
   - Replace *full_path* with the complete path and file name of the domain template jar file that you created when you ran the pack command to pack up the domain on the shared storage device.
   - Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the pack and unpack commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

5. Change directory to the newly created *IAD_MSERVER_HOME* directory and verify that the domain configuration files were copied to the correct location on the OAMHOST2 local storage device.

Starting the Node Manager in the Managed Server Domain Directory on OAMHOST2

Follow these steps to update and start the Node Manager from the Managed Server home:

1. Verify that the listen address in the nodemanager.properties file is set correctly, by completing the following steps:
   1. Change directory to the *IAD_MSERVER_HOME* binary directory:

      ```
      cd IAD_MSERVER_HOME/nodemanager
      ```

   2. Open the nodemanager.properties file for editing.
   3. Validate the `ListenAddress` property to the correct hostname as follows:

      ```
      OAMHOST2: ListenAddress=OAMHOST2
      ```

   4. Update the `ListenPort` property with the correct Listen Port details.
   5. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change directory to the *IAD_MSERVER_HOME* binary directory:

   ```
   cd IAD_MSERVER_HOME/bin
   ```

3. Use the following command to start the Node Manager:

   ```
   nohup ./startNodeManager.sh > $IAD_MSERVER_HOME/nodemanager/nodemanager.out 2>&1 &
   ```

For information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Removing OAM Server from WebLogic Server 12*c* defaultCoherenceCluster

You must exclude all Oracle Access Management (OAM) clusters (including policy manager and OAM runtime server) from the default WebLogic Server 12*c* coherence cluster using the WebLogic Server Administration Console.

From 12.2.1.3.0 onwards, OAM server-side session management uses database and does not require coherence cluster to be established. In some environments, warnings and errors are observed due to default coherence cluster initialized by WebLogic. To avoid or fix these errors, exclude all of the OAM clusters from default WebLogic Server coherence cluster using the following steps:

1. Log in to the WebLogic Server Administration Console, using the URL:

   ```
   http://IADADMINVHN.example.com:7001/console
   ```

2. In the left pane of the console, expand **Environment** and select **Coherence Clusters**.

   The Summary of Coherence Clusters page displays the Coherence cluster configurations that have been created in this domain.

3. Click **defaultCoherenceCluster** and select the **Members** tab.
4. Click **Lock and Edit**.
5. From **Servers and Clusters**, deselect all OAM clusters (including policy manager and OAM runtime server).
6. Click **Save**.
7. Click **Activate changes**.

Adding a Load Balancer Certificate to JDK Trust Stores

Some IAM Products require that the SSL certificate used by the load balancer be added to the trusted certificates in the JDK. To add the certificate, do the following:

1. Create a directory to hold user created keystores and certificates. For example:

   ```
   mkdir SHARED_CONFIG_DIR/keystores
   ```

2. Obtain the certificate from the load balancer. You can obtain the load balancer certificate from using a browser, such as Firefox. However, the easiest way to obtain the certificate is to use the **openssl** command. The syntax of the command is as follows:

   ```
   openssl s_client -connect LOADBALANCER -showcerts </dev/null 2>/dev/null|
   openssl x509 -outform PEM > SHARED_CONFIG_DIR/keystores/LOADBALANCER.pem
   ```

   For example:

   ```
   openssl s_client -connect login.example.com:443 -showcerts </dev/null
   2>/dev/null|openssl x509 -outform PEM > SHARED_CONFIG_DIR/keystores/
   login.example.com.pem
   ```

   > **Note:**
   >
   > This command saves the certificate to a file called `login.example.com.pem` in `SHARED_CONFIG_DIR/keystores`.

3. Load the certificate into the JDK and Node Manager Trust Stores by running the following command to import the CA certificate file, `login.example.com.pem`, into the JAVA_HOME.
4. Set **JAVA_HOME** to *JAVA_HOME*
5. Set **PATH** to include **JAVA_HOME/bin**.

6. Execute the following command to import the certificate into the Java trust store.

   ```
   keytool -importcert -alias login.example.com -file SHARED_CONFIG_DIR/
   keystores/login.example.com.pem -trustcacerts -keystore $JAVA_HOME/jre/lib/
   security/cacerts
   ```

7. Enter a password for the keystore. The default password for the JDK is *changeit*. The default password for the Node Manager keystores is *COMMON_IAM_PASSWORD*. You will be prompted to confirm that the certificate is valid.

Tuning the oamDS Data Source

For optimium performance, increase the number of connections allowed by the OAM data source.

To tune oamDS, complete the following steps:

1. Log in to the WebLogic Server Console at:

   ```
   http://IADADMINVHN.example.com:7001/console
   ```

2. Click **Lock & Edit**.
3. In **Domain Structure**, expand **Services**, and then click **Data Sources**.
4. Click **oamDS**.
5. In **Settings for oamDS**, click the **Configuration** tab, and then the **Connection Pool** tab. Change the values for the following:
   - **Initial Capacity** to 800
   - **Maximum Capacity** to 800
   - **Minimum Capacity** to 800
6. Click **Save**.
7. Click **Activate Changes**.

Enabling Virtualization

Use the Fusion Middleware Control to enable virtualization.

To enable virtualization:

1. Log in to Oracle Fusion Middleware console using the URL:

   ```
   http://IADADMINVHN.example.com:7001/em
   ```

2. Click **WebLogic Domain > Security > Security Provider Configuration**.
3. Expand **Security Store Provider**.
4. Expand **Identity Store Provider**.
5. Click **Configure**.
6. Add a custom property.
7. Select property “virtualize” with value “true” and click **OK**.
8. Click **OK** again to persist the change.

For more information about the virtualize property, see OPSS System and Configuration Properties in *Securing Applications with Oracle Platform Security Services*.

Configuring the WebLogic Proxy Plug-In

Before you can validate that requests are routed correctly through the Oracle HTTP Server instances, you must set the `WebLogic Plug-In Enabled` parameter. It is recommended to set the `WebLogic Plug-In Enabled` parameter at the domain level. Any clusters or servers not using the plugin via the web-tier can have their `WebLogic Plug-In Enabled` parameter value set to `no` on an exception basis as needed.

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Domain Structure** pane, click on the top-level domain node.
3. Click **Lock & Edit** in the Change Center.
4. Click on the Domain Name.
5. Click on the **Web Applications** tab.
6. Locate and select the **WebLogic PlugIn Enabled** option.
7. Click **Save**.
8. Click **Activate Changes** in the Change Center.
9. Restart the Administration Server.

14 Creating Infrastructure for Oracle Identity Governance

The following topics describe how to install and configure an initial domain, which can be used as the starting point for an enterprise deployment. Later chapters in this guide describe how to extend this initial domain with the various products and components that comprise the enterprise topology you are deploying.

A complete Oracle Identity and Access Management uses a split domain deployment, where there is a single domain for Oracle Access Management and a different one for Oracle Identity Governance. You must create a separate infrastructures for Access and Governance.

- Synchronizing the System Clocks
  Before you deploy Oracle Identity Governance, verify that the system clocks on each host computer are synchronized. You can do this by running the `date` command simultaneously on all the hosts in each cluster.
- About the Initial Infrastructure Domain
  Before you create the initial Infrastructure domain, ensure that you review the key concepts.
- Variables Used When Creating the Infrastructure Domain
  As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.
- Support for Dynamic Clusters in Infrastructure Domains
  Infrastructure domains support two different topologies: static clusters-based topology and dynamic clusters-based topology. When choosing the dynamic cluster topology, there are some differences with respect to the conventional static clusters configuration.
- Installing the Oracle Fusion Middleware Infrastructure on OIMHOST1
  Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for an enterprise deployment.
- Installing Oracle Identity Governance for an Enterprise Deployment
  The procedure for installing Oracle Identity Governance and the dependant softwares for an enterprise deployment is explained in this section.
- Creating the Database Schemas for Oracle Identity Governance
  Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.
- Configuring the Oracle Identity Governance Domain
  The following topics provide instructions for creating an Oracle Identity Governance domain using the Fusion Middleware Configuration wizard.
- Performing Additional Domain Configuration Steps
- Creating Oracle Identity Manager Authenticator
  Before you start the domain, you have to run a script which creates the Oracle Identity Manager (OIM) Authenticator in the domain.

- Configuring the Domain Directories and Starting the Servers
  After the domain is created and the Node Manager is configured, you can then configure the additional domain directories and start the Administration Server and any Managed Servers on the AdminHost.
- Configuring Listen Addresses When Using Dynamic Clusters
- Propagating the Domain and Starting the Servers on OIMHOST2
  After you start and validate the Administration Server and WLS_WSM1 Managed Server on OIMHOST1, you can then perform the following tasks on OIMHOST2.
- Modifying the Upload and Stage Directories to an Absolute Path
- About the Supported Authentication Providers
- Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group
  When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server authentication provider (`DefaultAuthenticator`). However, for an enterprise deployment, Oracle recommends that you use a dedicated, centralized LDAP-compliant authentication provider.
- Adding a Load Balancer Certificate to JDK Trust Stores for OIG
- Configuring the WebLogic Proxy Plug-In
- Backing Up the Configuration
  It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.
- Verification of Manual Failover of the Administration Server

Synchronizing the System Clocks

Before you deploy Oracle Identity Governance, verify that the system clocks on each host computer are synchronized. You can do this by running the `date` command simultaneously on all the hosts in each cluster.

Alternatively, there are third-party and open-source utilities you can use for this purpose.

About the Initial Infrastructure Domain

Before you create the initial Infrastructure domain, ensure that you review the key concepts.

- About the Infrastructure Distribution
- Characteristics of the Domain

About the Infrastructure Distribution

You create the initial Infrastructure domain for an enterprise deployment by using the Oracle Fusion Middleware Infrastructure distribution. This distribution contains both the Oracle WebLogic Server software and the Oracle JRF software.

The Oracle JRF software consists of Oracle Web Services Manager, Oracle Application Development Framework (Oracle ADF), Oracle Enterprise Manager Fusion Middleware Control, the Repository Creation Utility (RCU), and other libraries and technologies that are required to support the Oracle Fusion Middleware products.

> **Note:**
>
> The Access infrastructure does not use the Web Services Manager.

See Understanding Oracle Fusion Middleware Infrastructure in *Understanding Oracle Fusion Middleware*.

Characteristics of the Domain

The following table lists some of the key characteristics of the domain that you are about to create. Reviewing these characteristics helps you to understand the purpose and context of the procedures that are used to configure the domain.

Many of these characteristics are described in more detail in Understanding a Typical Enterprise Deployment.

Characteristic of the Domain	More Information
Uses a separate virtual IP (VIP) address for the Administration Server.	Configuration of the Administration Server and Managed Servers Domain Directories
Uses separate domain directories for the Administration Server and the Managed Servers in the domain.	Configuration of the Administration Server and Managed Servers Domain Directories
Uses a per domain Node Manager configuration.	About the Node Manager Configuration in a Typical Enterprise Deployment
Requires a separately installed LDAP-based authentication provider.	Understanding OPSS and Requests to the Authentication and Authorization Stores

Variables Used When Creating the Infrastructure Domain

As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.

These directory variables are defined in File System and Directory Variables Used in This Guide.

- ORACLE_HOME
- ASERVER_HOME
- MSERVER_HOME
- APPLICATION_HOME
- JAVA_HOME

> **Note:**
>
> To simplify, the above variables are used. Depending on the domain you are creating, you must add the prefix to the above variables with the infrastructure you are creating for. For example:
>
> - For access deployments, use IAD. For example: *IAD_ASERVER_HOME*
> - For governance, use IGD. For example: *IGD_ASERVER_HOME*

In addition, you reference the following virtual IP (VIP) addresses and host names that are defined in Physical and Virtual IP Addresses Required by the Enterprise Topology:

- ADMINVHN
- OIMHOST1
- OIMHOST2
- DBHOST1
- DBHOST2
- SCAN Address for the Oracle RAC Database (`DB-SCAN.example.com`)

> **Note:**
>
> Depending on the domain you are creating, you must add the prefix to ADMINVHN. For example, IAD_ADMINVHN.

> **Note:**
>
> The instructions in this section use the installation on OIMHOST1 and OIMHOST2 as an example. If you are creating the infrastructure domain for Access, then substitute OAMHOST1 and OAMHOST2 wherever appropriate.

Support for Dynamic Clusters in Infrastructure Domains

Infrastructure domains support two different topologies: static clusters-based topology and dynamic clusters-based topology. When choosing the dynamic cluster topology, there are some differences with respect to the conventional static clusters configuration.

Static clusters, also called configured clusters, are conventional clusters where you manually configure and add each server instance. A dynamic cluster includes a new "server-template" object that is used to define a centralized configuration for all generated (dynamic) server instances. When you create a dynamic cluster, the dynamic servers are preconfigured and automatically generated for you. This feature enables you to scale up the number of server instances in the 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.

The steps in this section include instructions to configure the domain for both static or dynamic topologies. The differences between the two types of configurations are listed below:

- The Configuration Wizard process may differ for each case. For example, you should define server templates for dynamic clusters instead of servers.
- For dynamic clusters, you should perform the server-specific configurations such as setting the listen address, configuring the upload and staging directories, or configuring the keystores in the server template instead of in the server.
- Service migration is configured in a different way for dynamic clusters. Dynamic clusters do not use migratable targets, instead the JMS resources are targeted to the cluster. Specific procedure for configuring service migration for dynamic clusters is included in this guide.

Mixed clusters (clusters that contains both dynamic and configured server instances) are not supported in the Oracle Identity and Access Management enterprise deployment.

Installing the Oracle Fusion Middleware Infrastructure on OIMHOST1

Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for an enterprise deployment.

- Installing a Supported JDK
- Starting the Infrastructure Installer
- Navigating the Infrastructure Installation Screens
- Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers
- Checking the Directory Structure
  After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.

Installing a Supported JDK

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system.

- Locating and Downloading the JDK Software
- Installing the JDK Software
  Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

Locating and Downloading the JDK Software

To find a certified JDK, see the certification document for your release on the Oracle Fusion Middleware Supported System Configurations page.

After you identify the Oracle JDK for the current Oracle Fusion Middleware release, you can download an Oracle JDK from the following location on Oracle Technology Network:

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the following locations:

- On the shared storage device, where it will be accessible from each of the application tier host computers, install the JDK in the location specified in File System and Directory Variables Used in This Guide.
- On the local storage device for each of the Web tier host computers. The Web tier host computers, which reside in the DMZ, do not necessarily have access to the shared storage on the application tier.
- On the local storage device for each of the directory tier host computers, in case of the directory hosts not utilizing the shared storage.

For more information about the recommended location for the JDK software, see Understanding the Recommended Directory Structure for an Enterprise Deployment.

To install JDK 1.8.0_211:

1. Change directory to the location where you downloaded the JDK archive file.

   ```
   cd download_dir
   ```

2. Unpack the archive into the JDK home directory, and then run the following commands:

   ```
   tar -xzvf jdk-8u201-linux-x64.tar.gz
   ```

   Note that the JDK version listed here was accurate at the time this document was published. For the latest supported JDK, see the *Oracle Fusion Middleware System Requirements and Specifications* for the current Oracle Fusion Middleware release.

3. Move the JDK directory to the recommended location in the directory structure.

   For example:

   ```
   mv ./jdk1.8.0_211 /u01/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

   For example:

   ```
   export JAVA_HOME=/u01/oracle/products/jdk
   export PATH=$JAVA_HOME/bin:$PATH
   ```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

   ```
   java -verison
   ```

   The Java version in the output should be displayed as “1.8.0_211”.

Starting the Infrastructure Installer

To start the installation program, perform the following steps.

1. Log in to OIMHOST1.

2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

   ```
   $JAVA_HOME/bin/java -d64 -jar distribution_file_name.jar
   ```

   In this example:

   - Replace *`JAVA_HOME`* with the environment variable or actual JDK location on your system.
   - Replace *`distribution_file_name`* with the actual name of the distribution JAR file.

     If you download the distribution from the Oracle Technology Network (OTN), then the JAR file is typically packaged inside a downloadable ZIP file.

     To install the software required for the initial Infrastructure domain, the distribution you want to install is:

     fmw_12.2.1.4.0_infrastructure_generic.jar.

     For more information about the actual file names of each distribution, see Identifying and Obtaining Software Downloads for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation. See Navigating the Installation Screens for a description of each installation program screen.

Navigating the Infrastructure Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name or click the **Help** button on the screen.

Table 14-1 Navigating the Infrastructure Installation Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears if you are installing any Oracle product on this host for the first time. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location. See Understanding the Oracle Central Inventory in *Installing Software with the Oracle Universal Installer*. **Note:** Oracle recommends that you configure the central inventory directory on the products shared volume. Example: `/u01/oracle/products/oraInventory` You may also need to execute the `createCentralinventory.sh` script as root from the `oraInventory` folder after the installer completes.
Welcome	This screen introduces you to the product installer.

Table 14-1 (Cont.) Navigating the Infrastructure Installation Screens

Screen	Description
Auto Updates	Use this screen to search My Oracle Support automatically for available patches or automatically search a local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For the purposes of an enterprise deployment, enter the value of the *IGD_ORACLE_HOME* variable listed in Table 8-2.
Installation Type	Use this screen to select the type of installation and as a consequence, the products and feature sets that you want to install. For this topology, select **Fusion Middleware Infrastructure**. **Note:** The topology in this document does not include server examples. Oracle strongly recommends that you do not install the examples into a production environment.
Prerequisite Checks	This screen verifies that your system meets the minimum requirements. If there are any warning or error messages, refer to the Oracle Fusion Middleware System Requirements and Specifications document on the Oracle Technology Network (OTN).
Security Updates	If you already have an Oracle Support account, use this screen to indicate how you would like to receive security updates. If you do not have one and are sure that you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.
Installation Summary	Use this screen to verify the installation options that you have selected. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file. Response files can be used later in a silent installation situation. For more information about silent or command-line installation, see Using the Oracle Universal Installer in Silent Mode in *Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click **Finish** to dismiss the installer.

Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers

If you have configured a separate shared storage volume or partition for secondary hosts, then you must install the Infrastructure on one of those hosts.

See Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

To install the software on the other host computers in the topology, log in to each host, and use the instructions in Starting the Infrastructure Installer and Navigating the Infrastructure Installation Screens to create the Oracle home on the appropriate storage device.

Checking the Directory Structure

After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.

To check the directory structure:

1. Change to the *ORACLE_HOME* directory where you installed the Infrastructure.
2. Enter the following command:

   ```
   ls --format=single-column
   ```

   The directory structure on your system must match the structure shown in the following example:

   ```
   cfgtoollogs
   coherence
   em
   inventory
   OPatch
   oracle_common
   oraInst.loc
   oui
   wlserver
   ```

   See What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

Installing Oracle Identity Governance for an Enterprise Deployment

The procedure for installing Oracle Identity Governance and the dependant softwares for an enterprise deployment is explained in this section.

- Starting the SOA Suite Installer on OIMHOST1
- Navigating the Oracle SOA Suite Installation Screens
- Starting the Oracle Identity and Access Management Installer
- Navigating the Oracle Identity and Access Management Installation Screens
- Verifying the Installation
- Downloading the Oracle Connector Bundle
  Download the Oracle Connector bundle using the instructions in this section.
- Installing the Oracle Identity Governance Connector
  After you download the Oracle Connector for LDAP, install it into the ORACLE_HOME directory.

Starting the SOA Suite Installer on OIMHOST1

To start the installation program:

1. Log in to OIMHOST1.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

   ```
   JAVA_HOME/bin/java -d64 -jar Installer File Name
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

   Replace *`fmw_12.2.1.4.0_soa_generic.jar`* with the name of the actual installer file for your product listed in Identifying and Obtaining Software Distributions for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation.

Navigating the Oracle SOA Suite Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name.

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you've already downloaded for your organization.
Installation Location	For Oracle Identity Governance enter *IGD_ORACLE_HOME*. For more information about Oracle Fusion Middleware directory structure, see Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.
Installation Type	Use this screen to select the type of installation and consequently, the products and feature sets you want to install. • Select **SOA Suite**
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the documents in the Roadmap for Verifying Your System Environment section in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
Installation Summary	Use this screen to verify the installation options you selected. Click **Install** to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click **Next** when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click **Finish** to dismiss the installer.

Starting the Oracle Identity and Access Management Installer

To start the installation program:

1. Log in to OIMHOST1.
2. Go to the directory where you downloaded the installation program.

3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

   ```
   JAVA_HOME/bin/java -d64 -jar fmw_12.2.1.4.0_idm_generic.jar
   ```

   Be sure to replace the JDK location in these examples with the actual JDK location on your system.

   Replace *`Installer File Name`* with the name of the actual installer file for your product listed in Identifying and Obtaining Software Distributions for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation.

Navigating the Oracle Identity and Access Management Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name.

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you've already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For Oracle Identity Governance, this must be set to *`IGD_ORACLE_HOME`*. For more information about Oracle Fusion Middleware directory structure, see Selecting Directories for Installation and Configuration in *Planning an Installation of Oracle Fusion Middleware*.
Installation Type	Use this screen to choose the type of installation you wish to deploy. You have two options: • Standalone Oracle Identity and Access Manager (Managed independently of Weblogic Server) Use this option if you are going to run Oracle Identity Governance with a webserver other than WebLogic. • Collocated Oracle Identity and Access Manager (Managed through WebLogic Server) Use this option if you have installed Oracle WebLogic Server into *`IGD_ORACLE_HOME`* as part of the infrastructure deployment. For Oracle Enterprise deployments, It is recommended that you use this option.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the documents in the Roadmap for Verifying Your System Environment section in *Installing and Configuring the Oracle Fusion Middleware Infrastructure*.
Installation Summary	Use this screen to verify the installation options you selected. Click **Install** to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click **Next** when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click **Finish** to dismiss the installer.

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- Reviewing the Installation Log Files
- Checking the Directory Structure
- Viewing the Contents of Your Oracle Home

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see Understanding Installation Log Files in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

The contents of your installation vary based on the options that you select during the installation.

The addition of Oracle Identity Governance adds the following directory and sub-directories. Use the `ls --format=single-column` command to verify the directory structure.

```
IGD_ORACLE_HOME/

OPatch
cfgtoollogs
coherence
em
idm
inventory
jdeveloper
mft
oep
oraInst.loc
oracle_common
osb
oui
soa
wlserver

idm/

clone
common
connectors
designconsole
idmdiag
idmtools
jlib
libovd
mbeans
modules
oam
oic
opam-connectors
plugins
remote_manager
```

```
schema
server
upgrade
```

For more information about the directory structure you should see after installation, see What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See Viewing the contents of an Oracle home in *Installing Software with the Oracle Universal Installer*.

Downloading the Oracle Connector Bundle

Download the Oracle Connector bundle using the instructions in this section.

Download the Oracle Connector bundle from the following location:

http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html

Copy the connector bundle for Oracle Internet Directory (it covers OUD as well) to the following directory:

```
IGD_ORACLE_HOME/idm/server/ConnectorDefaultDirectory
```

Installing the Oracle Identity Governance Connector

After you download the Oracle Connector for LDAP, install it into the ORACLE_HOME directory.

To do this perform the following steps:

1. Go to the following directory:

   ```
   cd IGD_ORACLE_HOME/idm/server/ConnectorDefaultDirectory
   ```

2. Unzip the LDAP directory using the following command:

   ```
   unzip oid_<version>.zip
   ```

Creating the Database Schemas for Oracle Identity Governance

Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.

- Oracle Identity Manager

  This automatically selects Oracle SOA Suite schemas along with the following ones:

  – Metadata Services (MDS)

  – Audit Services (IAU)

  – Audit Services Append (IAU_APPEND)

- Audit Services Viewer (IAU_VIEWER)
- Oracle Platform Security Services (OPSS)
- User Messaging Service (UMS)
- WebLogic Services (WLS)
- Common Infrastructure Services (STB)

Use the Repository Creation Utility (RCU) to create the schemas. This utility is installed in the Oracle home for each Oracle Fusion Middleware product. For more information about RCU and how the schemas are created and stored in the database, see Preparing for Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Complete the following steps to install the required schemas:

- Installing and Configuring a Certified Database
- Starting the Repository Creation Utility (RCU)
- Navigating the RCU Screens to Create the Schemas
- Verifying Schema Access
- Configuring OIM Schemas for Transactional Recovery

Installing and Configuring a Certified Database

Make sure that you have installed and configured a certified database, and that the database is up and running.

See the Preparing the Database for an Enterprise Deployment.

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Set the *JAVA_HOME* environment variable so it references the location where you installed a supported JDK.

   See File System and Directory Variables Used in This Guide.

2. Navigate to the following directory on OIMHOST1:

   ```
   IGD_ORACLE_HOME/oracle_common/bin
   ```

3. Start RCU:

   ```
   ./rcu
   ```

> **Note:**
>
> If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces created by the RCU, provide the -`encryptTablespace true` option when you start the RCU.
>
> This will default the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See Encrypting Tablespaces in *Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

Schema creation involves the following tasks:

Task 1 Introducing RCU
Review the Welcome screen and verify the version number for RCU. Click **Next** to begin.

Task 2 Selecting a Method of Schema Creation
If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have the necessary privileges.
If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option will generate a SQL script, which can be provided to your database administrator to create the required schema. See Understanding System Load and Product Load in *Creating Schemas with the Repository Creation Utility*.

Task 3 Providing Database Connection Details
Provide the database connection details for RCU to connect to your database.

1. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
2. Enter the **Port** number of the RAC database scan listener, for example 1521.
3. Enter the RAC **Service Name** of the database.
4. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.
5. Enter the **Password** of the user name that you provided in step 4.
6. If you have selected the SYS user, ensure that you set the role to SYSDBA.
7. Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

> **Tip:**
>
> For more information about the options on this screen, see Database Connection Details in *Creating Schemas with the Repository Creation Utility*.

Task 4 Specifying a Custom Prefix and Selecting Schemas

1. Specify the custom prefix you want to use to identify the Oracle Fusion Middleware schemas.

   The custom prefix is used to logically group these schemas together for use in this domain; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

> **Tip:**
>
> Make a note of the custom prefix you choose to enter here; you will need this later, during the domain creation process.
>
> For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.
>
> For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

2. Expand the group IDM Schemas, and then select the Oracle Identity Manager schema. All the relative schemas will be selected:
   - **Common infrastructure Services**
   - **Oracle Platform Security Services**
   - **User Messaging Service**
   - **Audit Services**
   - **Audit Services Append**
   - **Audit Services Viewer**
   - **Metadata Services**
   - **SOA Infrastructure**
   - **Weblogic Services**

There are two mandatory schemas that are selected by default. You cannot deselect them: **Common Infrastructure Services** (the STB schema) and **WebLogic Services** (the WLS schema). The **Common Infrastructure Services** schema enables you to retrieve information from RCU during domain configuration. See Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.

> **Tip:**
>
> For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Click **Next** to proceed, then click **OK** on the dialog window confirming that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords

Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database security requirements before you continue. RCU will proceed at this point even if you do not meet the password polices. Hence, perform this check outside RCU itself.

> **Tip:**
>
> You must make a note of the passwords you set on this screen; you will need them later on during the domain creation process.

Task 6 Verifying the Tablespaces for the Required Schemas
You can accept the default settings on the remaining screens, or you can customize how RCU creates and uses the required tablespaces for the Oracle Fusion Middleware schemas.

> **Note:**
>
> You can configure a Fusion Middleware component to use JDBC stores for JMS servers and Transaction Logs, by using the Configuration Wizard. These JDBC stores are placed in the Weblogic Services component tablespace. If your environment expects to have a high level of transactions and/or JMS activity, you can increase the default size of the <*PREFIX*>_WLS tablespace to better suit the environment load.

Click **Next** to continue, and then click **OK** on the dialog window to confirm the tablespace creation.
For more information about RCU and its features and concepts, see About the Repository Creation Utility in *Creating Schemas with the Repository Creation Utility*.

Task 7 Creating Schemas
Review the summary of the schemas to be loaded and click **Create** to complete schema creation.

> **Note:**
>
> If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and re-create the schemas before you continue.

Task 8 Reviewing Completion Summary and Completing RCU Execution
When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

```
sqlplus <RCU_PREFIX>_OIM/<PASSWORD>@//<SCAN_ADDRESS>:<PORT>/<SERVICE_NAME>
```

For example:

```
sqplus IGDEDG_OIM/<password>@//db-scan.example.com:1521/oigpdb_s.example.com
```

The output appears as follows:

```
SQL*Plus: Release 18.0.0.0.0 - Production on Mon Aug 9 01:53:57 2021
Version 18.5.0.0.0

Copyright (c) 1982, 2018, Oracle. All rights reserved.

Last Successful login time: Mon Aug 09 2021 01:52:44 -07:00

Connected to:
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production
Version 18.5.0.0.0

SQL>
```

Configuring OIM Schemas for Transactional Recovery

After you have installed the Oracle Identity Governance schemas successfully, use the procedure in this section to configure the schemas for transactional recovery.

This procedure sets the appropriate database privileges so that the Oracle WebLogic Server transaction manager can query the schemas for transaction state information and issue the appropriate commands, such as commit and rollback, during recovery of in-flight transactions after a WebLogic Server is unexpectedly unavailable.

These privileges should be granted to the owner of the OIM schema, which you defined when you created the schemas with the Repository Creation Utility.

To configure the OIM schemas for transactional recovery privileges:

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

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

2. Enter the following commands:

   ```
   SQL> Grant select on sys.dba_pending_transactions to oim_schema_prefix_oim;

   Grant succeeded.

   SQL> Grant force any transaction to oim_schema_prefix_oim;

   Grant succeeded.

   SQL>
   ```

Configuring the Oracle Identity Governance Domain

The following topics provide instructions for creating an Oracle Identity Governance domain using the Fusion Middleware Configuration wizard.

For more information on the other methods that are available for creating a domain, see Additional Tools for Creating, Extending, and Managing WebLogic Domains in *Creating WebLogic Domains Using the Configuration Wizard*.

- Starting the Configuration Wizard
- Navigating the Configuration Wizard Screens to Configure the Oracle Identity GovernanceDomain

Starting the Configuration Wizard

To begin domain configuration, run the following command in the Oracle Fusion Middleware Oracle home.

```
IAD_ORACLE_HOME/oracle_common/common/bin/config.sh
```

Navigating the Configuration Wizard Screens to Configure the Oracle Identity GovernanceDomain

Follow the instructions in the following sections to create and configure the domain for the topology, with static or dynamic clusters.

- Creating the Domain with Static Clusters
- Creating the Domain with Dynamic Clusters

Creating the Domain with Static Clusters

Follow the instructions in this section to create and configure the domain for the topology.

Domain creation and configuration includes the following tasks.

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a new domain**.
In the Domain Location field, specify the value of the *IGD_ASERVER_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen of the Configuration Wizard, see Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 2 Selecting the Configuration Templates
Select Oracle Identity Manager - 12.2.1.4.0[idm], the following template will be automatically selected:

- **Oracle Enterprise Manager - 12.2.1.4.0[em]**
- **Oracle WSM Policy Manager - 12.2.1.4.0[oracle_common]**
- **Oracle JRF - 12.2.1.4.0[oracle_common]**
- **WebLogic Coherence Cluster Extension - 12.2.1.4.0[wlserver]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 3 Configuring High Availability Options
This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply HA options (that is, the Configuration Wizard creates the JDBC stores and configures ASM for them).

> **Note:**
>
> Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster. So, the Configuration Wizard steps assume that the JDBC persistent stores are used along with Automatic Service Migration.
> If, for any reason, you want to use Files Stores, you can retain the default values for TLOGs and JMS persistent store options in this screen and configure them in a shared location later. See Configuring TLOGs File Persistent Store in a Shared Folder. Shared location is required to resume JMS and HA in a failover scenario.
> You can also configure TLOGs and JMS persistent stores manually in a post step.
> For information about the differences between JDBC and Files Stores, and for specific instructions to configure them manually, see JDBC Persistent Stores vs. File Persistent Stores.

On the High Availability Options screen:

- Select **Enable Automatic Service Migration** with **Database Leasing**.
- Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
- Set **JMS Server Persistence** to **JMS JDBC Store**.
- Click **Next**.

Task 4 Selecting the Application Home Location
On the Application Location screen, specify the value of the *APPLICATION_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the options on this screen can be found in Application Location in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 5 Configuring the Administrator Account
On the Administrator Account screen, specify the user name and password for the default WebLogic Administrator account for the domain.
Make a note of the user name and password specified on this screen; you will need these credentials later to boot and connect to the domain's Administration Server.
Click **Next**.

Task 6 Specifying the Domain Mode and JDK
On the Domain Mode and JDK screen:

- Select **Production** in the Domain Mode field.
- Select the **Oracle Hotspot** JDK in the JDK field.

Selecting **Production Mode** on this screen gives your environment a higher degree of security, requiring a user name and password to deploy applications and to start the Administration Server.

> **Tip:**
>
> More information about the options on this screen, including the differences between development mode and production mode, can be found in Domain Mode and JDK in *Creating WebLogic Domains Using the Configuration Wizard*.
> In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. See Creating the boot.properties File.

Click **Next**.

Task 7 Specifying the Database Configuration Type
On the Database Configuration Type screen:

- Select **RCU Data** to activate the fields on this screen.

  The **RCU Data** option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for the schemas needed to configure the domain.
- Verify that **Vendor** is `Oracle` and **Driver** is `*Oracle's Driver (Thin) for Service Connections; Versions: Any`.
- Verify that **Connection Parameters** is selected.

> **Note:**
>
> If you choose to select **Manual Configuration** on this screen, you will have to manually fill in the parameters for your schema on the JDBC Component Schema screen.

After you select **RCU Data**, fill in the fields as shown in the following table:

Field	Description
Host Name	Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the *Enterprise Deployment Workbook*. For information about the Enterprise Deployment Workbook, see Using the Enterprise Deployment Workbook.
DBMS/Service	Enter the service name for the Oracle RAC database appropriate for this domain where you will install the product schemas. For example: `iamedg.example.com` Specify the service name based on the value configured earlier in the Preparing the Database for an Enterprise Deployment section.
Port	Enter the port number on which the database listens. For example, `1521`.
Schema Owner Schema Password	Enter the user name and password for connecting to the database's Service Table schema. This is the schema user name and password that was specified for the Service Table component on the "Schema Passwords" screen in RCU (see Creating the Database Schemas). The default user name is *prefix*_STB, where *prefix* is the custom prefix that you defined in RCU.

Click **Get RCU Configuration** when you are finished specifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Click **Next** if the connection to the database is successful.

> **Tip:**
>
> More information about the **RCU Data** option can be found in Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*. More information about the other options on this screen can be found in Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Specifying JDBC Component Schema Information

Verify that the values on the JDBC Component Schema screen are correct for all schemas.

The schema table should be populated, because you selected **Get RCU Data** on the previous screen. As a result, the Configuration Wizard locates the database connection values for all the schemas required for this domain.

At this point, the values are configured to connect to a single-instance database. However, for an enterprise deployment, you should use a highly available Real Application Clusters (RAC) database, as described in Preparing the Database for an Enterprise Deployment.

In addition, Oracle recommends that you use an Active GridLink datasource for each of the component schemas. For more information about the advantages of using GridLink data sources to connect to a RAC database, see Database Considerations in the *High Availability Guide*.

To convert the data sources to GridLink:

1. Select all the schemas by selecting the checkbox at in the first header row of the schema table.
2. Click **Convert to GridLink** and click **Next**.

Task 9 Providing the GridLink Oracle RAC Database Connection Details

On the GridLink Oracle RAC Component Schema screen, provide the information required to connect to the RAC database and component schemas, as shown in following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, `1521`)
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the **Port** field, enter the ONS Remote port (typically, `6200`).
Enable Fan	Verify that the **Enable Fan** check box is selected, so the database can receive and process FAN events.

For more information about specifying the information on this screen, as well as information about how to identify the correct SCAN address, see Configuring Active GridLink Data Sources with Oracle RAC in the *High Availability Guide*.

You can also click **Help** to display a brief description of each field on the screen.

Click **Next**.

Task 10 Testing the JDBC Connections

Use the JDBC Component Schema Test screen to test the data source connections you have just configured.

A green check mark in the Status column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> More information about the other options on this screen can be found in Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*

Click **Next**.

Task 11 Entering Credentials
Enter the credentials you wish to use for the Oracle Identity Governance components. You have the choice of choosing both a username and a password for the various objects.

- **keystore**: Set the username to "keystore" and the password to the password you wish to use for all automatically created keystores.
- **OIMSchemaPassword**: Set the username to the OIM schema which you created in the earlier sections. For example, `IGD_OIM` (username) and its associated password.
- **Sysadmin**: This is the administrative user you will use for OIM. This is typically `xelsysadm`, but can be anything. Set the password to a value you wish to use for this account.
- **WebLogicAdminKey**: This is the domain admin username and password. For example, `weblogic`.

Click **Next**.

Task 12 Keystore
Use this screen to specify details about the keystore to be used in the domain.
For a typical enterprise deployment, you can leave the default values.
See Keystore in *Creating WebLogic Domains Using the Configuration Wizard*.
Click **Next**.

Task 13 Selecting Advanced Configuration

> **Note:**
>
> This is not required for Access infrastructure.

To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Administration Server**

  This is required to properly configure the listen address of the Administration Server.
- **Node Manager**

  This is required to configure Node Manager.
- **Topology**

  This is required to add, delete, or modify the Settings for Server Templates, Managed Servers, Clusters, Virtual Targets, and Coherence.

- **Domain Frontend Host Capture**:

  This allows you to specify the public entry point for OIM.

> **Note:**
>
> When using the Advanced Configuration screen in the Configuration Wizard, if any of the above options are not available on the screen, then return to the Templates screen, and be sure you selected the required templates for this topology.

Click **Next**.

Task 14 Configuring the Administration Server Listen Address
On the Administration Server screen:

1. In the **Server Name** field, retain the default value - AdminServer.
2. In the **Listen Address** field, enter the virtual host name that corresponds to the VIP of the ADMINVHN that you procured in Procuring Resources for an Enterprise Deployment and enabled in Preparing the Host Computers for an Enterprise Deployment.

   For more information on the reasons for using the ADMINVHN virtual host, see Reserving the Required IP Addresses for an Enterprise Deployment.
3. In the **Listen Port** field, enter the port number to access the administration server. This guide recommends you to use the default port `7101` for Governance.

   Leave the other fields at their default values. In particular, be sure that no server groups are assigned to the Administration Server.

Click **Next**.

Task 15 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type, then specify the following Node Manager credentials you will use to connect to the Node Manager:

- Username: This is the user name used to connect to the Node Manager. For example, `admin`.
- Password and Confirm Password: Enter the password you wish to associate with the Node Manager username.

> **Tip:**
>
> For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.
> For more information about per domain and per host Node Manager implementations, see About the Node Manager Configuration in a Typical Enterprise Deployment.
> For information about Node Manager configurations, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Click **Next**.

Task 16 Configuring Managed Servers
Use the Managed Servers screen to create two new Managed Servers:

1. Click the **Add** button to create a new Managed Server.
2. Specify `WLS_WSM1` in the **Server name** column.
3. In the **Listen Address** column, enter OIMHOST1.

   Be sure to enter the host name that corresponds to OIMHOST1; do not use the IP address.
4. In the **Listen Port** column, enter `WSM_PORT - 7010`.
5. In the **Server Groups** drop-down list, select **JRF-MAN-SVR** and **WSMPM-MAN-SVR**.

   These server groups ensure that the Oracle JRF and Oracle Web Services Manager (OWSM) services are targeted to the Managed Servers that you are creating.

   Server groups target Fusion Middleware applications and services to one or more servers by mapping defined groups of application services to each defined server group. Any application services that are mapped to a given server group are automatically targeted to all servers that are assigned to that group. See Application Service Groups, Server Groups, and Application Service Mappings in *Domain Template Reference*.

   > **Note:**
   >
   > Nonce caching for Oracle Web Services is initialized automatically by the WSM-CACHE-SVR server group and is suitable for most custom applications. This initialization is automatically performed in SOA, OSB, and other FMW servers that run JRF and create a coherence cluster. Nonce is a unique number that can be used only once in a SOAP request and is used to prevent replay attacks. Nonce caching naturally scales with the number of added Managed Servers that run Web service applications.
   >
   > For information about advanced caching configurations, see Caching the Nonce with Oracle Coherence in *Securing Web Services and Managing Policies with Oracle Web Services Manager*, which provides additional guidance for the use of nonce caching and the WSM-CACHE-SVR server-group in custom WLS servers.
6. Repeat this process to create a second Managed Server named `WLS_WSM2`.

   For the **Listen Address**, enter *OIMHOST2*. For the **Listen Port**, enter 7010. Apply the same server groups that you applied to the first managed server to the WLS_WSM2.

The Managed Server names suggested in this procedure (WLS_WSM1 and WLS_WSM2) are referenced throughout this document; if you choose different names then be sure to replace them as needed.
On the Managed Servers screen, a new Managed Server for Oracle SOA Suite and Oracle Identity Manger appears in the list of servers. This server was created automatically by the Oracle SOA Suite configuration template you selected in Table 8-3.
Perform the following tasks to modify the default Oracle SOA Suite and Oracle Identity Manager Managed Server and create a second Managed Server:

1. Rename the default Managed Server to `oim_server1` to `WLS_OIM1`.

   Rename the default Managed Server `soa_server1` to `WLS_SOA1`.
2. Click **Add** to create a new Oracle Identity Governance Managed Server, and name it `WLS_OIM2`.

> **Tip:**
>
> The server names recommended here will be used throughout this document; if you choose different names, be sure to replace them as needed.

3. Click **Add** to create a new Oracle SOA Suite Managed Server, and name it `WLS_SOA2`.
4. Use the information in Oracle Identity Governance Managed Server Details to fill in the rest of the columns for each Oracle Identity Governance Managed Server.

For more information about the options on the Managed Server screen, see Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Listen Port	Enable SSL	SSL Listen Port	Server Groups
WLS_OIM1	OIMHOST1	14000	No	Disabled	OIM-MGD-SVRS
WLS_OIM2	OIMHOST2	14000	No	Disabled	OIM-MGD-SVRS
WLS_SOA1	OIMHOST1	8001	No	Disabled	SOA-MGD-SVRS
WLS_SOA2	OIMHOST2	8001	No	Disabled	SOA-MGD-SVRS

Click **Next**.

Task 17 Configuring a Cluster
In this task, you create a cluster for each set of Managed Servers. You can then target the Oracle Identity Governance and Oracle SOA Suite components to the relevant cluster. Create the following clusters:

- OIM_Cluster
- SOA_Cluster
- WSM-PM_Cluster

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify the cluster name in the **Cluster Name** field.
3. Repeat the steps to create all of the clusters.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Administering Clusters for Oracle WebLogic Server*.

Click **Next**.
For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Assigning Server Templates
Click **Next** .

Task 19 Configuring Dynamic Servers

Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

> **Note:**
>
> This screen will not be displayed if you are creating the Access Infrastructure.

Task 20 Assigning Managed Servers to the Cluster

Use the Assign Servers to Clusters screen to assign your managed servers to the clusters you have just created. At the end of this you will have the following assignments:

Cluster	Managed Servers
OIM_Cluster	WLS_OIM1 WLS_OIM2
SOA_Cluster	WLS_SOA1 WLS_SOA2
WSM-PM_Cluster	WLS_WSM1 WLS_WSM2

1. In the Clusters pane, select the cluster to which you want to assign the servers.
2. In the Servers pane, assign the Managed Servers to the cluster as in the table above, using one of the following methods:
   - Click on the Managed Server to select it, and then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
   - Double-click on the Managed Server to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign each of the Managed Server to the respective cluster.
4. Click **Next**.

For more information about the options on this screen, see Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 21 Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain.
In the **Cluster Listen Port**, enter `9991`.

> **Note:**
>
> For Coherence licensing information, Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Click **Next**.

Task 22 Creating Machines
Use the Machines screen to create new machines in the domain. A machine is required in order for the Node Manager to be able to start and stop the servers.
You must create a machine even if your topology contains just the Administration Server.

1. Select the **Unix Machine** tab.
2. Click the **Add** button to create new UNIX machines.

   Use the values in Values to Use When Creating Unix Machines to define the Name and Node Manager Listen Address of each machine.
3. Verify the port in the Node Manager Listen Port field.

   The port number `5556`, shown in this example, may be referenced by other examples in the documentation. Replace this port number with your own port number as needed.

Name	Node Manager Listen Address	Node Manager Listen Port
ADMINHOST	Enter the value of the ADMINVHN variable.	5556
OIMHOST1	The value of the OIMHOST1 host name variable or OIMHOST1 alias. For example, `OIMHOST1.example.com`.	5556
OIMHOST2	The value of the OIMHOST2 host name variable or OIMHOST2 alias. For example, `OIMHOST2.example.com`.	5556

> **Note:**
>
> If you are installing OIM on the same host as Oracle Access Management (OAM), ensure that the Node Manager ports are unique to each deployment.

> **Tip:**
>
> More information about the options on this screen can be found in Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 23 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign the Oracle Identity Governance Managed Servers to the corresponding machines in the domain.
Assign the machines as shown in the following table:

Servers	Machines
AdminHost	AdminServer
WLS_OIM1 WLS_SOA1 WLS_WSM1	OIMHOST1
WLS_OIM2 WLS_SOA2 WLS_WSM2	OIMHOST2

1. In the Machines pane, select the machine to which you want to assign the servers.
2. In the Servers pane, assign the Managed Servers to the machine as in the table above, using one of the following methods:
   - Click on the Managed Server to select it, and then click on the right arrow to move it beneath the selected machines in the machines pane.
   - Double-click on the Managed Server to move it beneath the selected machine in the machines pane.
3. Repeat to assign each of the Managed Server to the respective machine.
4. Click **Next**.

For more information about the options on this screen, see Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 24 Creating Virtual Targets
Click **Next**.

Task 25 Creating Partitions
Click **Next**.

Task 26 Configuring Domain Front End Host
In the Domain Front End host screen you specify the main entry point for OIM. This will equate to the name on the load balancer. For example, set **Plain** to `http://prov.example.com`.

> **Note:**
>
> Even though you are specifying this value it will never be used.

SSL: https://prov.example.com
Default: SSL
Click **Next**.

Task 27 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation will not begin until you click **Create**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 28 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

..

Creating the Domain with Dynamic Clusters

Follow the instructions in this section to create and configure the domain for the topology.

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a new domain**.
In the Domain Location field, specify the value of the *ASERVER_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the other options on this screen of the Configuration Wizard, see Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 2 Selecting the Configuration Templates
On the Templates screen, make sure that **Create Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Identity Manager - 12.2.1.4.0[oim]**
- **Oracle Enterprise Manager - 12.2.1.4.0[em]**

  Selecting this template automatically selects the following dependencies:

  - **Oracle JRF - 12.2.1.4.0[oracle_common]**
  - **WebLogic Coherence Cluster Extension - 12.2.1.4.0[wlserver]**
- **Oracle WSM Policy Manager - 12.2.1.4.0[oracle_common]**

> **Tip:**
>
> More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 3 Configuring High Availability Options
This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply these HA options.
On the High Availability Options screen, complete the following steps:

1. Verify that **Enable Automatic Service Migration** is not selected.
2. Verify that Default Persistent Store is selected as the **JTA Transaction Log Persistence** option.
3. Select JDBC Store as the **JMS Service Persistence** option.

You can configure only JMS Server persistence for Dynamic Clusters by using the Configuration Wizard. You cannot configure Service Migration and JTA Transaction Logs Persistence for Dynamic Clusters by using the Configuration Wizard, you have to configure them manually. Instructions are covered in later chapters of this guide.
Click **Next**.

Task 4 Selecting the Application Home Location

On the Application Location screen, specify the value of the *APPLICATION_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

> **Tip:**
>
> More information about the options on this screen can be found in Application Location in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 5 Configuring the Administrator Account

On the Administrator Account screen, specify the user name and password for the default WebLogic Administrator account for the domain.
Make a note of the user name and password specified on this screen; you need to use these credentials later to boot and connect to the Administration Server domain.
Click **Next**.

Task 6 Specifying the Domain Mode and JDK

On the Domain Mode and JDK screen:

- Select only **Production** in the Domain Mode field.
- Select the **Oracle Hotspot** JDK in the JDK field.

Selecting **Production Mode** on this screen gives your environment a higher degree of security, requiring a user name and password to deploy applications and to start the Administration Server.

> **Tip:**
>
> More information about the options on this screen, including the differences between development mode and production mode, can be found in Domain Mode and JDK in *Creating WebLogic Domains Using the Configuration Wizard*.
> In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. See Creating the boot.properties File.

Click **Next**.

Task 7 Specifying the Database Configuration Type

On the Database Configuration Type screen:

- Select **RCU Data** to activate the fields on this screen.

  The **RCU Data** option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema. This connection automatically retrieves schema information for the schemas to configure the domain.
- Verify that **Vendor** is `Oracle` and **Driver** is `*Oracle's Driver (Thin) for Service Connections; Versions: Any`.
- Verify that **Connection Parameters** is selected.

> **Note:**
>
> If you choose to select **Manual Configuration** on this screen, you have to manually fill in the parameters for the schema on the JDBC Component Schema screen.

After you select **RCU Data**, fill in the fields as shown in the following table:

Field	Description
Host Name	Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the *Enterprise Deployment Workbook*.
DBMS/Service	Enter the service name for the Oracle RAC database where you will install the product schemas. For example: `orcl.example.com` Be sure to specify the common service name that is used to identify all the instances in the Oracle RAC database; do not use the host-specific service name.
Port	Enter the port number on which the database listens. For example, `1521`.
Schema Owner Schema Password	Enter the user name and password for connecting to the database's Service Table schema. The schema user name and password that was specified for the Service Table component on the "Schema Passwords" screen in RCU (see Creating the Database Schemas) is used here. The default user name is *`prefix`*`_STB`, where *`prefix`* is the custom prefix that you defined in RCU.

Click **Get RCU Configuration** when you are finished specifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Click **Next** if the connection to the database is successful.

> **Tip:**
>
> More information about the **RCU Data** option can be found in Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*. More information about the other options on this screen can be found in Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*

Task 8 Specifying JDBC Component Schema Information

Verify that the values on the JDBC Component Schema screen are correct for all schemas. The schema table is populated because you selected **Get RCU Data** on the previous screen. As a result, the Configuration Wizard locates the database connection values for all the schemas required for this domain.

At this point, the values are configured to connect to a single-instance database. However, for an enterprise deployment, you must use a highly available Real Application Clusters (RAC) database, as described in Preparing the Database for an Enterprise Deployment.

In addition, Oracle recommends that you use an Active GridLink datasource for each of the component schemas. For more information about the advantages of using GridLink data sources to connect to a RAC database, see Database Considerations in the*High Availability Guide*.

To convert the data sources to GridLink:

1. Select all the schemas by selecting the checkbox at in the first header row of the schema table.
2. Click **Convert to GridLink** and click **Next**.

Task 9 Providing the GridLink Oracle RAC Database Connection Details

On the GridLink Oracle RAC Component Schema screen, provide the information required to connect to the RAC database and component schemas, as shown in following table.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, `1521`)
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the **Port** field, enter the ONS Remote port (typically, `6200`).
Enable Fan	Verify that the **Enable Fan** check box is selected, so the database can receive and process FAN events.

For more information about specifying the information on this screen, as well as information about how to identify the correct SCAN address, see Configuring Active GridLink Data Sources with Oracle RAC in the *High Availability Guide*.

You can also click **Help** to display a brief description of each field on the screen.

Click **Next**.

Task 10 Testing the JDBC Connections

Use the JDBC Component Schema Test screen to test the data source connections you have configured.

A green check mark in the Status column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> More information about the other options on this screen can be found in Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*

Click **Next**.

Task 11 Entering Credentials
Enter the credentials you wish to use for the Oracle Identity Governance components. You have the choice of choosing both a username and a password for the various objects.

- **keystore**: Set the username to keystore and the password to the password you wish to use for all automatically created keystores.
- **OIMSchemaPassword**: Set the username to the OIM schema which you created in the earlier sections. For example, `IGD_OIM` (username) and its associated password.
- **Sysadmin**: This is the administrative user you will use for OIM. This is typically `xelsysadm`, but can be anything. Set the password to a value you wish to use for this account.
- **WebLogicAdminKey**: This is the domain admin username and password. For example, `weblogic`.

Click **Next**.

Task 12 Keystore
Use this screen to specify details about the keystore to be used in the domain.
For a typical enterprise deployment, you can leave the default values.
See Keystore in *Creating WebLogic Domains Using the Configuration Wizard*.
Click **Next**.

Task 13 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Administration Server**

  This is required to configure the listen address of the Administration Server.
- **Node Manager**

  This is required to configure Node Manager.
- **Topology**

  This is required to add, delete, or modify the Settings for Server Templates, Managed Servers, Clusters, Virtual Targets, and Coherence.

> **Note:**
>
> When using the Advanced Configuration screen in the Configuration Wizard:
>
> - If any of the options are not available on the screen, then return to the Templates screen, and ensure that you have selected the required templates for this topology.
> - Do not select the **Domain Frontend Host Capture** advanced configuration option. Later, you have to configure the frontend host property for specific clusters, rather than for the domain.

Click **Next**.

Task 14 Configuring the Administration Server Listen Address
On the Administration Server screen:

1. In the **Server Name** field, retain the default value: `AdminServer`.
2. In the **Listen Address** field, enter the virtual host name that corresponds to the VIP of the ADMINVHN that you procured in Procuring Resources for an Enterprise Deployment and enabled in Preparing the Host Computers for an Enterprise Deployment.

   For more information on the reasons for using the ADMINVHN virtual host, see Reserving the Required IP Addresses for an Enterprise Deployment.
3. In the **Listen Port** field, enter the port number to access the administration server. This guide recommends you to use the default port 7001 for Access and 7101 for Governance.

   Leave the other fields at their default values. In particular, be sure that no server groups are assigned to the Administration Server.

Click **Next**.

Task 15 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type, then specify the following Node Manager credentials you will use to connect to the Node Manager:

- Username: This is the user name used to connect to the Node Manager. For example, `admin`.
- Password and Confirm Password: Enter the password you wish to associate with the Node Manager username.

> **Tip:**
>
> For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.
> For more information about per domain and per host Node Manager implementations, see About the Node Manager Configuration in a Typical Enterprise Deployment.
> For additional information, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Click **Next**.

Task 16 Configuring Managed Servers

On the Managed Servers screen, a new Managed Server for Oracle Identity Governance appears in the list of servers. These servers were created automatically by the Oracle Identity Governance configuration template you selected in File System and Directory Variables Used in This Guide.

Static Managed Server definitions are not needed for dynamic cluster configurations. To remove the default Managed Servers, complete the following steps:

1. Click on the Managed Server.
2. Click **Delete**.
3. Repeat for each of the Managed Servers.
4. Click **Next**.

Task 17 Configuring a Cluster

In this task, you create a cluster of Managed Servers to which you can target the Oracle Identity Governance software.

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `OIM_Cluster` in the **Cluster Name** field.
3. From the **Dynamic Server Groups** drop-down list, select `OIM-DYN-CLUSTER`.
4. Create a second cluster called `SOA_Cluster` and assign the Dynamic Server group `SOA-DYN-CLUSTER`.
5. Create a third cluster called **WSM-PM_Cluster** and assign the Dynamic Server group **WSMPM-DYN-CLUSTER**.

> **Note:**
>
> By default, server instances in a cluster communicate with one another using unicast. If you want to change your cluster communications to use multicast, refer to Considerations for Choosing Unicast or Multicast in *Administering Clusters for Oracle WebLogic Server*.

> **Tip:**
>
> More information about the options on this screen can be found in Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

> **Tips:**
>
> For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 18 Assigning Server Templates

Use the Server Templates screen to configure the template:

1. Verify that `wsmpm-server-template` is selected in the **Name** field.
2. Specify `7009` in the **Listen Port** field.
3. Leave the **Enable SSL** option unchecked.
4. Verify `OIM-server-template` is listed in the **Name** field.
5. Specify `13999` in the **Listen Port** field.
6. Leave the **Enable SSL** option unchecked.
7. Specify `8000` for the **Listen Port** for template soa-server-template.
8. Click **Next** .

Task 19 Configuring Dynamic Servers
Use the Dynamic Clusters screen to configure the following clusters:

Cluster Name	Server Name Prefix	Server Template	Dynamic Cluster Size	Machine Name Match Expression	Calculated Machine Names
oim_cluster	WLS_OIM	oim-server-template	2	OIMHOST*	Selected
soa_cluster	WLS_SOA	soa-server-template	2	OIMHOST*	Selected
WSM-PM_Cluster	WLS_WSM	wsmpm-server-template	2	OIMHOST*	Selected

Complete the following steps on this screen:

1. Verify `OIM_Cluster` is listed in the **Cluster Name** field.
2. Specify `WLS_OIM` in the **Server Name Prefix** field.
3. From the **Server Template** drop-down list, select `OIM-server-template`.
4. Specify `2` in the **Dynamic Server Count** field.
5. Specify `OIMHOST*` in the **Machine Name Match Expression** field.
6. Select **Calculated Machine Names**, **Calculated Listen Ports**, and **Dynamic Cluster fields**.

> **Note:**
>
> Dynamic clusters with the Calculated Listen Port option selected will have incremental port numbers for each dynamic managed server that is created automatically: dynamic server 1 will use Listen Port+1, dynamic server 2 will use Listen Port+2.
>
> Since the Listen Port configured is `13999` and calculated ports is checked, OIM dynamic servers will use the following:
>
> - WLS_OIM1:14000
> - WLS_OIM2:14001
>
> Since the SOA Listen Port configured is `8000` and calculated ports is checked, SOA dynamic servers will use the following:
>
> - WLS_SOA1:8001
> - WLS_SOA2:8002
>
> Since the Listen Port that is configured is 7009 and calculated ports is checked, WSMPM dynamic servers use the following ports:
>
> - WLS_WSM1: 7010
> - WLS_WSM2: 7011

7. Repeat the steps 1 through 6 for each of the clusters to be created.
8. Click **Next**.

> **Note:**
>
> The Configuration Wizard does not allow you to specify a specific listen address for dynamic servers. For information about setting a specific listen address for WebLogic servers that are members of a dynamic cluster, see Configuring Listen Addresses in Dynamic Cluster Server Templates.

Task 20 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain.
In the **Cluster Listen Port**, enter `9991`.

> **Note:**
>
> For Coherence licensing information, Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Click **Next**.

Task 21 Creating Machines
Use the Machines screen to create new machines in the domain. A machine is required in order for the Node Manager to be able to start and stop the servers.

1. Select the **Unix Machine** tab.
2. Click the **Add** button to create the new UNIX machines.

   Use the values in Table 14-3 to define the Name and Node Manager Listen Address of each machine.
3. Verify the port in the Node Manager Listen Port field.

   The port number `5556`, shown in this example, may be referenced by other examples in the documentation. Replace this port number with your own port number as needed.

Name	Node Manager Listen Address	Node Manager Listen Port
ADMINHOST	Enter the value of the ADMINVHN variable.	5556
OIMHOST1	The value of the OIMHOST1 host name variable or OIMHOST1 alias. For example, `OIMHOST1.example.com`.	5556
OIMHOST2	The value of the OIMHOST2 host name variable or OIMHOST2 alias. For example, `OIMHOST2.example.com`.	5556

> **Note:**
>
> The name of the machine should reflect the value that you have specified in the **Machine Match Expression** field with the addition of a sequential number. That is, if you have specified `OIMHOST*` in the **Machine Match Expression** field, then the names of your machines should be OIMHOST1, OIMHOST2, and so on.

> **Tip:**
>
> More information about the options on this screen can be found in Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 22 Assigning Servers to Machines
Use the Assign Servers to Machines screen to assign any statically defined managed servers to the appropriate machines. Servers that are part of a dynamic cluster are assigned to the calculated machine names automatically.
Assign AdminServer to the ADMINHOST machine.
Click **Next**.

Task 23 Creating Virtual Targets
Click **Next**.

Task 24 Creating Partitions
Click **Next**.

Task 25 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.

Click **Update** to execute the domain extension.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 26 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation begins when you click **Create**.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Click **Next**.

Task 27 Writing Down Your Domain Home and Administration Server URL
The Configuration Success screen shows the following items about the domain you have configured:

- Domain Location
- Administration Server URL

You must make a note of both items because you need them later; the domain location is required to access the scripts that are used to start the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Performing Additional Domain Configuration Steps

Use the Configuration Wizard to update the newly created domain. Perform the following steps:

1. Restart the Configuration Wizard by running the following command:

   ```
   IGD_ORACLE_HOME/oracle_common/common/bin/config.sh
   ```

2. On the Configuration Type screen, select **Update an existing domain**.
3. In the Domain Location field, specify the Domain home directory (*IGD_ASERVER_HOME*).
4. Click **Next**.
5. On the Templates screen, select **Update Domain Using Custom Template**.
6. In the Template location field, specify `IGD_ORACLE_HOME/soa/common/templates/wls/oracle.soa.classic.domain_template.jar`.
7. Click **Next**.

8. On the GridLink Oracle RAC Component Schema screen, click **Next**.
9. On the JDBC Component Test Schema Test screen, click **Next**.
10. On the Advanced Configuration screen, click **Next**.
11. On the Configuration Summary screen, click **Update**.
12. After the domain is extended, click **Next** on the Configuration Progress screen, and click **Finish** on the End of Configuration screen.

After you have completed creating the domain with static clusters, go to Creating Oracle Identity Manager Authenticator.

Creating Oracle Identity Manager Authenticator

Before you start the domain, you have to run a script which creates the Oracle Identity Manager (OIM) Authenticator in the domain.

To do this, complete the following steps:

1. Set the `DOMAIN_HOME` to *`IGD_ASERVER_HOME`* using the following command:

   ```
   export DOMAIN_HOME=IGD_ASERVER_HOME
   ```

2. Run the following command from the location *`IGD_ORACLE_HOME`*`/idm/server/bin`:

   ```
   ./offlineConfigManager.sh
   ```

   > **Note:**
   >
   > If you do not have execute permissions for this file, add it using the following command:
   >
   > ```
   > chmod 750 offlineConfigManager.sh
   > ```

Configuring the Domain Directories and Starting the Servers

After the domain is created and the Node Manager is configured, you can then configure the additional domain directories and start the Administration Server and any Managed Servers on the AdminHost.

- Starting the Node Manager in the Administration Server Domain Home
  Use these steps to start the per-domain Node Manager for the *IAD_ASERVER_HOME* domain directory.
- Creating the boot.properties File
  You must create a `boot.properties` if you want to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.
- Disabling the Derby Database

- Enabling the Managed Servers to use IPv6 Networking
  If the Managed Server is configured to use IPv6 networking, then you may encounter issues when you start the Managed Server.
- Setting the Memory Parameters in IAMGovernanceDomain
- Starting the Administration Server Using the Node Manager
  After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.
- Validating the Administration Server
  Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control; both of these are installed and configured on the Administration Servers.
- Creating a Separate Domain Directory for Managed Servers
  When you initially create the domain for enterprise deployment, the domain directory resides on a shared disk. This default domain directory is used to run the Administration Server. You can now create a copy of the domain on the local storage for each of your managed server hosts. The domain directory on the local (or private) storage is used to run the Managed Servers.
- Starting the Node Manager in the Managed Server Domain Directory on OIMHOST1
- Configuring Listen Addresses When Using Dynamic Clusters
- Starting and Validating the WLS_WSM1 Managed Server on OIMHOST1
  After you have configured Node Manager and created the Managed Server domain directory, you can use WebLogic Administration Console to start the WLS_WSM1 Managed Server on OIMHOST1.

Starting the Node Manager in the Administration Server Domain Home

Use these steps to start the per-domain Node Manager for the *IAD_ASERVER_HOME* domain directory.

1. Verify that the listen address in the `nodemanager.properties` file is set correctly.
   a. Open the nodemanager.properties file for editing:

      ```
      vi IAD_ASERVER_HOME/nodemanager/nodemanager.properties
      ```

   b. Make sure the `ListenAddress` property is set to the value of the ADMINVHN virtual IP address.
   c. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change to the following directory:

   ```
   cd IAD_ASERVER_HOME/bin
   ```

3. Start the Node Manager by entering the following command:

   ```
   nohup ./startNodeManager.sh > IAD_ASERVER_HOME/nodemanager/nodemanager.out 2>&1 &
   ```

For more information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Creating the boot.properties File

You must create a `boot.properties` if you want to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.

To create a `boot.properties` file for the Administration Server:

1. Create the following directory structure:

   ```
   mkdir -p IAD_ASERVER_HOME/servers/AdminServer/security
   ```

2. In a text editor, create a file called `boot.properties` in the `security` directory that you created in the previous step, and enter the Administration Server credentials that you defined when you ran the Configuration Wizard to create the domain:

   ```
   username=adminuser
   password=password
   ```

   > **Note:**
   >
   > When you start the Administration Server, the `username` and `password` entries in the file are encrypted.
   >
   > For security reasons, minimize the amount of time the entries in the file are left unencrypted; after you edit the file, you should start the server as soon as possible so that the entries are encrypted.

3. Save the file and close the editor.

Disabling the Derby Database

Disable the embedded Derby database, which is a file-based database, packaged with Oracle WebLogic Server. The Derby database is used primarily for development environments. As a result, you must disable it when you are configuring a production-ready enterprise deployment environment; otherwise, the Derby database process starts automatically when you start the Managed Servers.

To disable the Derby database:

1. Navigate to the following directory in the Oracle home:

   ```
   cd WL_HOME/common/derby/lib
   ```

2. Rename the Derby library jar file:

   ```
   mv derby.jar disable_derby.jar
   ```

3. If each host uses a separate file system, repeat steps 1 and 2 on each host.

Enabling the Managed Servers to use IPv6 Networking

If the Managed Server is configured to use IPv6 networking, then you may encounter issues when you start the Managed Server.

To do this, complete the following steps:

1. Edit the *IAD_ASERVER_HOME*/bin/setUserOverrides.sh file to add the following line:

   ```
   JAVA_OPTIONS="${JAVA_OPTIONS} -Djava.net.preferIPv4Stack=true"
   ```

   > **Note:**
   >
   > If the file does not exist, then create it.

2. Save and close the file.

Setting the Memory Parameters in IAMGovernanceDomain

The initial startup parameter in the IAMGovernanceDomain, which defines the memory usage, is insufficient for production systems. If you are using the deployment for production purposes it is recommended that you increase the value of this parameter.

The example below sets the minimum heap size to 4GB and the maximium heap size to 8GB. To change the memory allocation setting, do the following:

1. Change the following memory allocation in the ASERVER_HOME/bin/setUserOverrides.sh file, by updating the Java maximum memory allocation pool (Xmx) to 3072m and initial memory allocation pool (Xms) to 1024m. For example, change the following line to be:

   ```
   MEM_ARGS="-Xms4096m -Xmx8192m"
   ```

2. Click **Save** and close the file.

Starting the Administration Server Using the Node Manager

After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.

To start the Administration Server by using the Node Manager:

1. Start the WebLogic Scripting Tool (WLST):

   ```
   cd ORACLE_COMMON_HOME/common/bin
   ./wlst.sh
   ```

2. Connect to Node Manager by using the Node Manager credentials:

   ```
   wls:/offline>nmConnect('nodemanager_username','nodemanager_password',
                'ADMINVHN','5556','domain_name',
                'IAD_ASERVER_HOME')
   ```

> **Note:**
>
> This user name and password are used only to authenticate connections between Node Manager and clients. They are independent of the server administrator ID and password and are stored in the `nm_password.properties` file located in the following directory:
>
> ```
> IAD_ASERVER_HOME/config/nodemanager
> ```

3. Start the Administration Server:

   ```
   nmStart('AdminServer')
   ```

   > **Note:**
   >
   > When you start the Administration Server, it attempts to connect to Oracle Web Services Manager for WebServices policies. It is expected that the WSM-PM Managed Servers are not yet started, and so, the following message appears in the Administration Server log:
   >
   > ```
   > <Warning><oracle.wsm.resources.policymanager>
   > <WSM-02141><Unable to connect to the policy access service due to
   > Oracle WSM policy manager host server being down.>
   > ```

4. Exit WLST:

   ```
   exit()
   ```

Validating the Administration Server

Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control; both of these are installed and configured on the Administration Servers.

To navigate to the Oracle WebLogic Server Administration Console, enter the following URL, and log in with the Oracle WebLogic Server administrator credentials :

```
http://IGDADMINVHN.example.com:7101/console
```

Creating a Separate Domain Directory for Managed Servers

When you initially create the domain for enterprise deployment, the domain directory resides on a shared disk. This default domain directory is used to run the Administration Server. You can now create a copy of the domain on the local storage for each of your managed server hosts. The domain directory on the local (or private) storage is used to run the Managed Servers.

> **Note:**
>
> If you are creating a domain for Oracle Access Management, it is not necessary to perform this step at this time. This is because, at the time of infrastructure creation, there are no managed servers in existence yet.

Placing the *IGD_MSERVER_HOME* on local storage is recommended to eliminate the potential contention and overhead caused by servers writing logs to shared storage. It is also faster to load classes and jars need from the domain directory, so any temporary or cache data that the Managed Servers use from the domain directory is processed quicker.

As described in Preparing the File System for an Enterprise Deployment, the path to the Administration Server domain home is represented by the *IGD_ASERVER_HOME* variable, and the path to the Managed Server domain home is represented by the *IGD_MSERVER_HOME* variable.

To create the Managed Server domain directory:

1. Sign in to the host running the Administration Server, for example, OIMHOST1, and run the `pack` command to create a template as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./pack.sh -managed=true \
             -domain=IGD_ASERVER_HOME \
             -template=/full_path/edgdomaintemplate.jar \
             -template_name=edg_domain_template \
         -log_priority=DEBUG \
             -log=/tmp/pack.log
   ```

   In this example:

   - Replace *IGD_ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.
   - Replace *full_path* with the complete path to the location where you want to create the domain template jar file. You need to reference this location when you copy or unpack the domain template jar file. It is recommended to choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

     You must specify a full path for the template jar file as part of the `-template` argument to the pack command:

     ```
     SHARED_CONFIG_DIR/domains/template_filename.jar
     ```

   - The `edgdomaintemplate.jar` file is a sample name for the jar file that you create, which contains the domain configuration files.
   - The `edg_domain_template` label is the label is assigned to the template data stored in the template file.

2. Make a note of the location of the `edgdomaintemplate.jar` file that you just created with the pack command.

Tip:

For more information about the `pack` and `unpack` commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

3. If you have not already, create the recommended directory structure for the Managed Server domain on the OIMHOST1 local storage device.

   Use the examples in File System and Directory Variables Used in This Guide as a guide.

4. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

```
cd ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=IGD_MSERVER_HOME \
            -overwrite_domain=true \
            -template=/full_path/edgdomaintemplate.jar \
        -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME
```

Note:

The `-overwrite_domain` option in the `unpack` command allows you to unpack a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this unpack operation.

Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional JAVA command-line options for running the servers, or specify additional environment variables. Any customizations that you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when you use the `pack` and `unpack` commands.

In this example:

- Replace *IGD_MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain is unpacked.
- Replace `/full_path/edgdomaintemplate.jar` with the complete path and file name of the domain template jar file that you created when you ran the pack command to pack the domain on the shared storage device.
- Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the `pack` and `unpack` commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

5. Change directory to the newly created Managed Server directory and verify that the domain configuration files were copied to the correct location on the OIMHOST1 local storage device.

Starting the Node Manager in the Managed Server Domain Directory on OIMHOST1

After you create the Managed Server domain directory, there are two domain home directories and two corresponding Node Manager instances on OIMHOST1. You use one Node Manager to control the Administration Server, running from Administration Server domain home, and you use the other Node Manager to control the Managed Servers, running from the Managed Server domain home.

You must start the two Node Managers independently.

> **Note:**
>
> The Node Manager for the Managed Server's *MSERVER_HOME* will be reset every time the domain configuration is unpacked. The `ListenAddress` will be changed to the *ADMINVHN* instead of the correct hostname. This needs to be changed to the correct value before starting the Node Manager service after an unpack is performed.

Follow these steps to update and start the Node Manager from the Managed Server home:

1. Verify that the listen address in the nodemanager.properties file is set correctly, by completing the following steps:
   a. Change to the following directory:

      ```
      IGD_MSERVER_HOME/nodemanager/
      ```

   b. Open the nodemanager.properties file for editing.
   c. Update the `ListenAddress` property to the correct hostname as follows:

      ```
      OIMHOST1: ListenAddress=OIMHOST1
      ```

   d. Update the `ListenPort` property with the correct Listen Port details.
   e. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change to the following directory:

   ```
   IGD_MSERVER_HOME/bin
   ```

3. Use the following command to start the Node Manager:

   ```
   nohup ./startNodeManager.sh > IGD_MSERVER_HOME/nodemanager/nodemanager.out 2>&1 &
   ```

For information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Configuring Listen Addresses When Using Dynamic Clusters

The default configuration for dynamic managed servers in dynamic clusters is to listen on all available network interfaces. In most cases, the default configuration may be undesirable. To limit the listen address to a specific address when you use dynamic clusters, see Configuring Listen Addresses in Dynamic Cluster Server Templates. Reverify the test URLs that are provided in the previous sections after you change the listen address and restart the clustered managed servers.

Starting and Validating the WLS_WSM1 Managed Server on OIMHOST1

After you have configured Node Manager and created the Managed Server domain directory, you can use WebLogic Administration Console to start the WLS_WSM1 Managed Server on OIMHOST1.

1. Enter the following URL into a browser to display the WebLogic Administration Console:

   ```
   http://IGDADMINVHN:7101/console
   ```

   In this example:

   - Replace *IGDADMINVHN* with the host name assigned to the IGDADMINVHN Virtual IP address in Identifying and Obtaining Software Downloads for an Enterprise Deployment.
   - Port `7101` is the typical port used for the Administration Server console and Fusion Middleware Control. However, you should use the actual URL that was displayed at the end of the Configuration Wizard session when you created the domain.

   > **Note:**
   >
   > The default Admin Port number for the Identity Domain is `7101`.

2. Sign-in to the WebLogic Administration Console by using the administrator's account. For example: `weblogic`.
3. In the Domain Structure window, expand the **Environment** node, then select **Servers**.
4. From the Summary of Servers page, click the **Control** tab.
5. Select **WLS_WSM1** from the Servers column of the table and click **Start**.
6. In the resulting Server Life Cycle Assistant window, click **Yes** to acknowledge the start of the server.
7. To verify that the Managed Server is working correctly, open your browser and enter the following URL:

   ```
   http://OIMHOST1.example.com:7010/wsm-pm
   ```

8. Enter the domain admin user name and password when prompted.

> **Note:**
>
> Use the port number appropriately, as assigned for your static or dynamic cluster. If you select the Calculate Listen Port option for dynamic clusters, the port number for each dynamic managed server that is automatically created is incremented by one: dynamic server 1 will use Listen Port+1, dynamic server 2 will use Listen Port+2.
>
> Since the Listen Port configured for Dynamic Cluster is 7009 and calculated ports is checked, WSMPM dynamic servers use the following ports:
>
> - *http://OIMHOST1*:*7010*/wsm-pm/
> - *http://OIMHOST2*:*7011*/wsm-pm/

Configuring Listen Addresses When Using Dynamic Clusters

The default configuration for dynamic managed servers in dynamic clusters is to listen on all available network interfaces. In most cases, the default configuration may be undesirable. To limit the listen address to a specific address when you use dynamic clusters, see Configuring Listen Addresses in Dynamic Cluster Server Templates. Reverify the test URLs that are provided in the previous sections after you change the listen address and restart the clustered managed servers.

Propagating the Domain and Starting the Servers on OIMHOST2

After you start and validate the Administration Server and WLS_WSM1 Managed Server on OIMHOST1, you can then perform the following tasks on OIMHOST2.

- Unpacking the Domain Configuration on OIMHOST2
- Starting the Node Manager in the Managed Server Domain Directory OIMHOST2
- Starting and Validating the WLS_WSM2 Managed Server on OIMHOST2
  After you have configured Node Manager and created the Managed Server domain directory, you can use WebLogic Administration Console to start the WLS_WSM2 Managed Server on OIMHOST2

Unpacking the Domain Configuration on OIMHOST2

Now that you have the Administration Server and the first WLS_WSM1 Managed Server running on OIMHOST1, you can configure the domain on OIMHOST2.

1. Log in to OIMHOST2.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the OIMHOST2 storage device.

   Use the examples in File System and Directory Variables Used in This Guide as a guide.
3. Make sure the `oimdomaintemplate.jar` accessible to OIMHOST2.

   For example, if you are using a separate shared storage volume or partition for OIMHOST2, then copy the template to the volume or partition mounted to OIMHOST2.

4. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./unpack.sh -domain=IGD_MSERVER_HOME
               -overwrite_domain=true
               -template=/full_path/create_domain.jar
               -log_priority=DEBUG
               -log=/tmp/unpack.log
               -app_dir=APPLICATION_HOME
   ```

   In this example:

   - Replace *IGD_MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
   - Replace *full_path* with the complete path and file name of the domain template jar file that you created when you ran the pack command to pack up the domain on the shared storage device.
   - Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

   > **Tip:**
   >
   > For more information about the pack and unpack commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

5. Change directory to the newly created *IGD_MSERVER_HOME* directory and verify that the domain configuration files were copied to the correct location on the OIMHOST2 local storage device.

Starting the Node Manager in the Managed Server Domain Directory OIMHOST2

Follow these steps to update and start the Node Manager from the Managed Server home:

1. Verify that the listen address in the `nodemanager.properties` file is set correctly, by completing the following steps:

   a. Change directory to the *IGD_MSERVER_HOME*/nodemanager directory:

      ```
      cd IGD_MSERVER_HOME/nodemanager
      ```

   b. Open the nodemanager.properties file for editing.

   c. Validate the `ListenAddress` property to the correct hostname as follows:

      ```
      OIMHOST2: ListenAddress=OIMHOST2
      ```

   d. Update the `ListenPort` property with the correct Listen Port details.

   e. Make sure that QuitEnabled is set to ‘true’. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change directory to the *MSERVER_HOME* binary directory:

   ```
   cd MSERVER_HOME/bin
   ```

3. Use the following command to start the Node Manager:

   ```
   nohup ./startNodeManager.sh > $IGD_MSERVER_HOME/nodemanager/nodemanager.out 2>&1 &
   ```

For information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

Starting and Validating the WLS_WSM2 Managed Server on OIMHOST2

After you have configured Node Manager and created the Managed Server domain directory, you can use WebLogic Administration Console to start the WLS_WSM2 Managed Server on OIMHOST2

1. Enter the following URL into a browser to display the WebLogic Administration Console:

   ```
   http://IGDADMINVHN:7101/console
   ```

   In this example:

   - Replace *IGDADMINVHN* with the host name assigned to the IGDADMINVHN Virtual IP address in Identifying and Obtaining Software Downloads for an Enterprise Deployment.
   - Port `7101` is the typical port used for the Administration Server console and Fusion Middleware Control. However, you should use the actual URL that was displayed at the end of the Configuration Wizard session when you created the domain.

   > **Note:**
   >
   > The default Admin Port number for the Identity Domain is `7101`.

2. Sign-in to the WebLogic Administration Console by using the administrator's account. For example: `weblogic`.
3. In the Domain Structure window, expand the **Environment** node, then select **Servers**.
4. From the Summary of Servers page, click the **Control** tab.
5. Select **WLS_WSM2** from the Servers column of the table and click **Start**.
6. In the resulting Server Life Cycle Assistant window, click **Yes** to acknowledge the start of the server.
7. To verify that the Managed Server is working correctly, open your browser and enter the following URL:

   ```
   http://OIMHOST2.example.com:7010/wsm-pm
   ```

8. Enter the domain admin user name and password when prompted.

> **Note:**
>
> Use the port number appropriately, as assigned for your static or dynamic cluster. If you select the Calculate Listen Port option for dynamic clusters, the port number for each dynamic managed server that is automatically created is incremented by one: dynamic server 1 will use Listen Port+1, dynamic server 2 will use Listen Port+2.
>
> Since the Listen Port configured for Dynamic Cluster is 7009 and calculated ports is checked, WSMPM dynamic servers use the following ports:
>
> - *http://OIMHOST1*:*7010*/wsm-pm/
> - *http://OIMHOST2*:*7011*/wsm-pm/

Modifying the Upload and Stage Directories to an Absolute Path

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. See Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment.

About the Supported Authentication Providers

Oracle Fusion Middleware supports a variety of LDAP authentication providers. See Identity Store Types and WebLogic Authenticators in *Securing Applications with Oracle Platform Security Services*.

The instructions in this guide assume that you are using Oracle Unified Directory.

> **Note:**
>
> By default, the instructions here describe how to configure the identity service instance to support querying against a single LDAP identity store with an unencrypted connection.
>
> If the connection to your identity provider has to be secured through SSL, then additional keystore configuration is required for role management in the Enterprise Manager Fusion Middleware Control to function correctly. For additional configuration information, see Doc ID 1670789.1.
>
> For more information about configuring a Multi-LDAP lookup, refer to Configuring the Identity Store Service in *Securing Applications with Oracle Platform Security Services*.

Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group

When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server authentication provider (`DefaultAuthenticator`). However, for an enterprise deployment, Oracle recommends that you use a dedicated, centralized LDAP-compliant authentication provider.

The following topics describe how to use the Oracle WebLogic Server Administration Console to create a new authentication provider for the enterprise deployment domain. This procedure assumes that you have already installed and configured a supported LDAP directory, such as Oracle Unified Directory or .

- About the Enterprise Deployment Users and Groups
- Creating the New Authentication Provider
- Deleting OIMSignatureAuthenticator
- Recreating OUDAuthenticator
- Adding the Administration Role to the New Administration Group
- Updating the boot.properties File and Restarting the System

About the Enterprise Deployment Users and Groups

The following topics provide important information on the purpose and characteristics of the enterprise deployment administration users and groups.

- About Using Unique Administration Users for Each Domain
- About the Domain Connector User
- About Adding Users to the Central LDAP Directory
- About Product-Specific Roles and Groups for Oracle Identity and Access Management
- Example Users and Groups Used in This Guide

About Using Unique Administration Users for Each Domain

When you use a central LDAP user store, you can provision users and groups for use with multiple Oracle WebLogic Server domains. As a result, there is a possibility that one WebLogic administration user can have access to all the domains within an enterprise.

It is a best practice to create and assign a unique distinguished name (DN) within the directory tree for the users and groups that you provision for the administration of your Oracle Fusion Middleware domains.

For example, create two users called `oamLDAP` and `oimLDAP` which is used to connect the WebLogic domain to LDAP. This allows the domain to see the users and groups which exist in the directory. You can create a different user for each domain or use a single user for multiple domains. Under no circumstances should the default LDAP administration user be used for this purpose. You must create these users in the `systemids` container. This container is used for system users that are not normally visible to users. Placing the user into the `systemids` container ensures that customers who have Oracle Identity Governance do not reconcile this user.

Using a different user for Oracle Access Management (OAM) and Oracle Identity Manager (OIM) LDAP connections ensures that the user that OAM uses to connect to LDAP has a restricted privilege set.

Create a user called `weblogic_iam` and an administration group called `WLSAdministrators`. Users in the `WLSAdministrators` group will be allowed to access the following:

- Oracle Fusion Middleware Control

- Oracle WebLogic Administration Console

Create a user called `oamadmin` and an administration group called `OAMAdministrators`. Users in the `OAMAdministrators` group are allowed to access the following:

- Oracle Access Policy Manager
- Oracle Access Manager Console

About the Domain Connector User

Oracle recommends that you create a separate domain connector user (for example, `oimLDAP`) in your LDAP directory. This user allows the domain to connect to the LDAP directory for the purposes of user authentication. It is recommended that this user be a non-administrative user.

In a typical Oracle Identity and Access Management deployment, you create this user in the `systemids` container. This container is used for system users that are not normally visible to users. Placing the user into the `systemids` container ensures that customers who have Oracle Identity Governance do not reconcile this user.

About Adding Users to the Central LDAP Directory

After you configure a central LDAP directory to be the authenticator for the enterprise domain, then you should add all new users to the new authenticator and not to the default WebLogic Server authenticator.

Users are added to the directory using the `idmConfigTool`. See Preparing an Existing LDAP Directory.

When you are using multiple authenticators (a requirement for an enterprise deployment), login and authentication will work, but role retrieval will not. The role is retrieved from the first authenticator only. If you want to retrieve roles using any other authenticator, then you must enable virtualization for the domain.

About Product-Specific Roles and Groups for Oracle Identity and Access Management

Each Oracle Fusion Middleware product implements its own predefined roles and groups for administration and monitoring.

Oracle Identity and Access Management has a number of groups that can be used to define who can access each of the products in the suite. The typical roles include:

- Accessing Weblogic or Fusion Middleware consoles
- Accessing Oracle Access Manager Components
- Accessing Oracle Identity Manager components

You can create different groups for each type. However, in this guide, we will be using the following (you can choose your own names:

- Weblogic Administrators: Example group name — `WLSAdministrators`
- OIM Administrators: Example group name — `OIMAdministrators`

For instructions on adding additional roles to the `WLSAdministrators` group, see Common Configuration and Management Tasks for an Enterprise Deployment.

Example Users and Groups Used in This Guide

These users will be created as a part of Preparing an Existing LDAP Directory.

- Admin User DN:

  ```
  cn=weblogic_iam,cn=users,dc=example,dc=com
  ```

- Admin Group DN:

  ```
  cn=WLSAdministrators,cn=groups,dc=example,dc=com
  ```

- Product-specific LDAP Connector User:

  ```
  cn=oimLDAP,cn=systemids,dc=example,dc=com
  ```

  This is the user that you use to connect WebLogic Managed Servers to the LDAP authentication provider. This user must have permissions to read and write to the Directory Trees:

  ```
  cn=users,dc=example,dc=com
  cn=groups,dc=example,dc=com

  cn=reserve,dc=example,dc=com
  ```

> **Note:**
>
> This user needs to be granted membership in the following groups to provide read and write access:
>
> ```
> cn=orclFAUserReadPrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAUserWritePrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAGroupReadPrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAGroupWritePrivilegeGroup,cn=groups,dc=example,dc=com
> ```

Creating the New Authentication Provider

After creating the new domain, if you are using LDAP and want to log in using LDAP, then you must create an authentication provider for the directory inside the OIG domain.

To create a new LDAP-based authentication provider:

1. Change directory to `IGD_ORACLE_HOME/idm/server/ssointg/config`.
2. Edit the `configureWLSAuthnProviders.config` file as shown below:

   ```
   OIM_WLSHOST: IGDADMINVHN.example.com
   OIM_WLSPORT: 7101
   OIM_WLSADMIN: weblogic
   OIM_WLSADMIN_PWD: <password>
   IDSTORE_DIRECTORYTYPE: OUD
   IDSTORE_HOST: idstore.example.com
   IDSTORE_PORT: 1389
   IDSTORE_BINDDN: cn=oudadmin
   IDSTORE_BINDDN_PWD: <password>
   ```

```
IDSTORE_USERSEARCHBASE: cn=Users,dc=example,dc=com
IDSTORE_GROUPSEARCHBASE: cn=Groups,dc=example,dc=com
```

3. Save the file.

Table 14-4 Properties of the configureWLSAuthnProviders.config File

Attribute	Description
OIM_WLSHOST	It is the listen address of the `IAMGovernanceDomain` Administration Server. For example: `IGDADMINVHN`.
OIM_WLSPORT	It is the Administration Server's port. For example: `7101`.
OIM_WLSADMIN	It is the name of the administration user for the `IAMGovernanceDomain`.
OIM_WLSADMIN_PWD	It is the password of the WLSADMIN account. The password is optional. If you do not provide the password, you will be prompted for it.
IDSTORE_DIRECTORYTYPE	It is the type of the LDAP directory you are using. For example: OID or OUD.
IDSTORE_HOST	It is the load balancer name for the LDAP directory. For example: `idstore.example.com`.
IDSTORE_PORT	It is the LDAP port on the load balancer. For example: 1389 for OUD.
IDSTORE_BINDDN	It is the credential used to connect to the directory to perform administrative actions. For example: `oudadmin` for OUD.
IDSTORE_BINDDN_PWD	This is the password for BINDDN. This is option if you do not provide it you will be prompted for it. The password is optional. If you do not provide the password, you will be prompted for it.
IDSTORE_USERSEARCHBASE	It is the location in the directory where user details are stored.
IDSTORE_GROUPSEARCHBASE	It is the location in the directory where group details are stored.

4. Execute the `OIGOAMIntegration.sh` script for creating the authenticator. For example:

```
cd IGD_ORACLE_HOME/idm/server/ssointg/bin
export JAVA_HOME=JAVA_HOME
export ORACLE_HOME=IGD_ORACLE_HOME
export WL_HOME=IGD_ORACLE_HOME/wlserver
./OIGOAMIntegration.sh -configureWLSAuthnProviders
```

5. Verify that there are no errors.

> **Note:**
>
> If this is the first time you are using OIGOAMIntegration.sh, you may need to mark it as an executable by using the following commands:
>
> ```
> chmod 750 ORACLE_HOME/idm/server/ssointg/bin/OIGOAMIntegration.sh
> chmod 750 ORACLE_HOME/idm/server/ssointg/bin/_OIGOAMIntegration.sh
> ```

Deleting OIMSignatureAuthenticator

The createWLSAuthenticator script creates a new security provider called OIMSignatureAuthenticator. This security provider is not required in Oracle Identity Manager 12c.

To delete the security provider:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. Click **Security Realms** on the left navigation pane.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. Select the security provider **OIMSignatureAuthenticator**.
7. Click **Delete**.
8. Click **Yes** to confirm the deletion.
9. Click **Activate Changes** to propagate the changes.

Recreating OUDAuthenticator

If your target directory is OUD, then you must delete and recreate the OUDAuthenticator security provider.

To delete the security provider:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. Click **Security Realms** on the left navigation pane.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. Select the security provider **OUDAuthenticator**.
7. Click **Delete**.
8. Click **Yes** to confirm the deletion.
9. Click **Activate Changes** to propagate the changes.

To recreate the security provider:

1. Log in to the WebLogic Server Administration Console using the URL.

`http:/IGDADMIN.example.com:7101/console`

2. Click **Security Realms** in the left navigational bar.
3. Click the **myrealm** default realm entry.
4. Click the **Providers** tab.
5. Click **Lock & Edit** in the Change Center.
6. Click the **New** button below the **Authentication Providers** table.
7. Enter a name for the provider.

   Use one of the following names, based on the LDAP directory service you are planning to use as your credential store:

   `OUDAuthenticator` for Oracle Unified Directory
8. From the **Type** drop-down list, select the authenticator type **OracleUnifiedDirectoryAuthenticator** for Oracle Unified Directory.
9. Click **OK** to return to the Providers screen.
10. On the Providers screen, click the newly created authenticator in the table.
11. Select **SUFFICIENT** from the **Control Flag** drop-down menu.

    Setting the control flag to **SUFFICIENT** indicates that if the authenticator can successfully authenticate a user, then the authenticator should accept that authentication and should not continue to invoke any additional authenticators.

    If the authentication fails, it will fall through to the next authenticator in the chain. Make sure all subsequent authenticators also have their control flags set to **SUFFICIENT**; in particular, check the **`DefaultAuthenticator`** and make sure that its control flag is set to **SUFFICIENT**.
12. Click **Save** to persist the change of the control flag setting.
13. Click the **Provider Specific** tab and enter the details specific to your LDAP server, as shown in the following table.

> **Note:**
>
> Only the required fields are discussed in this procedure. For information about all the fields on this page, consider the following resources:
>
> - To display a description of each field, click **Help** on the **Provider Specific** tab.
> - For more information on setting the **User Base DN**, **User From Name Filter**, and **User Attribute** fields, see Configuring Users and Groups in the Oracle Internet Directory and Oracle Virtual Directory Authentication Providers in *Administering Security for Oracle WebLogic Server*.

Parameter	Sample Value	Value Description
Host	For example: `idstore.example.com`	The LDAP server's server ID.
Port	For example: `1389`	The LDAP server's port number.

Parameter	Sample Value	Value Description
Principal	For example: `cn=oimLDAP,cn=systemids,dc=example,dc=com`	The LDAP user DN used to connect to the LDAP server.
Credential	Enter LDAP password.	The password used to connect to the LDAP server.
SSL Enabled	Unchecked (clear)	Specifies whether SSL protocol is used when connecting to the LDAP server.
User Base DN	For example: `cn=users,dc=example,dc=com`	Specify the DN under which your users start.
All Users Filter	`(&(uid=*)(objectclass=person))`	Instead of a default search criteria for **All Users Filter**, search all users based on the `uid` value. If the **User Name Attribute** for the user object class in the LDAP directory structure is a type other than `uid`, then change that type in the **User From Name Filter** field. For example, if the **User Name Attribute** type is `cn`, then this field should be set to: `(&(cn=*)(objectclass=person)))`
User From Name Filter	For example: `(&(uid=%u)(objectclass=person))`	If the **User Name Attribute** for the user object class in the LDAP directory structure is a type other than `uid`, then change that type in the settings for the **User From Name** Filter. For example, if the **User Name Attribute** type is `cn`, then this field should be set to: `(&(cn=%u)(objectclass=person)))`.
User Name Attribute	For example: `uid`	The attribute of an LDAP user object that specifies the name of the user.
Use Retrieved User Name as Principal	Checked	Must be turned on.
Group Base DN	For example: `cn=groups,dc=example,dc=com`	Specify the DN that points to your Groups node.
All Groups Filter	`(&(cn=*)(objectclass=groupOfUniqueNames))`	
GUID Attribute	`entryuuid`	This value is prepopulated with `entryuuid` when `OracleUnifiedDirectoryAuthenticator` is used for OUD. Check this value if you are using Oracle Unified Directory as your authentication provider.

14. Click **Save** to save the changes.

15. Return to the Providers page by clicking **Security Realms** in the right navigation pane, clicking the default realm name (**myrealm**), and then **Providers**.

16. Click **Reorder** and use the resulting page to reorder the list of providers so that they match the order given below:

    List of Authentication Providers

    - OAMIDAsserter
    - OUDAuthenticator

- DefaultAuthenticator
- OIMAuthenticationProvider
- Trust Service Identity Asserter
- DefaultIdentityAsserter

17. Click **OK**.
18. In the Change Center, click **Activate Changes**.
19. Restart the domain using the following commands:

    a. Shut down the Managed Servers WLS_OIM1 and WLS_OIM2.

    b. Shut down the Managed Servers WLS_SOA1 and WLS_SOA2.

    c. Shut down the Managed Servers WLS_WSM1 and WLS_WSM2.

    d. Shut down the Administration Server.

    e. Restart the Administration Server.

    f. Start the Managed Servers WLS_SOA1 and WLS_SOA2.

    g. Start the Managed Servers WLS_OIM1 and WLS_OIM2.

    h. Start the Managed Servers WLS_WSM1 and WLS_WSM2.

    If you have performed the workaround as described in the Update Value of MatchLDAPAttribute in oam-config.xml, then you must also restart the OAM domain.

    Shut down and restart the Administration Server and all the Managed Servers (WLS_AMA1, WLS_AMA2, WLS_OAM1, WLS_OAM2).

Adding the Administration Role to the New Administration Group

After you have added the users and groups to LDAP, The WLSAdministrators group must be assigned to the Administration role within the Weblogic domain security realm.This enables all users that belong to the group to be administrators for the domain.

To assign the Administration role to the new enterprise deployment administration group:

1. Log in to the WebLogic Administration Server Console by using the administration credentials that you provided in the Configuration Wizard.

   Do not use the credentials for the administration user that you created and provided for the new authentication provider.
2. In the left pane of the Administration Console, click **Security Realms**.
3. Click the default security realm (**myrealm**).
4. Click the **Roles and Policies** tab.
5. Expand the **Global Roles** entry in the table and click **Roles**.
6. Click the **Admin** role.
7. Click **Add conditions**.
8. Select **Group** from the **Predicate List** drop-down menu, and then click **Next**.
9. Enter `WLSAdministrators` in the **Group Argument Name** field, and then click **Add**.

   `WLSAdministrators` is added to the list box of arguments.
10. Click **Finish** to return to the Edit Global Role page.

The `WLSAdministrators` group is now listed.

11. Click **Save** to finish adding the **Admin** Role to the `WLSAdministrators` group.

12. Validate that the changes were made by logging in to the WebLogic Administration Server Console by using the new `weblogic_iam` user credentials.

    If you can log in to the Oracle WebLogic Server Administration Console and Fusion Middleware Control with the credentials of the new administration user that you just provisioned in the new authentication provider, then you have configured the provider successfully.

Updating the boot.properties File and Restarting the System

> **Note:**
>
> This task is not required for Access Infrastructure.

After you create the new administration user and group, you must update the Administration Server `boot.properties` file with the administration user credentials that you created in the LDAP directory:

1. On OIMHOST1, go to the following directory:

   ```
   IGD_ASERVER_HOME/servers/AdminServer/security
   ```

2. Rename the existing `boot.properties` file:

   ```
   mv boot.properties boot.properties.backup
   ```

3. Use a text editor to create a file called `boot.properties` under the security directory.

4. Enter the following lines in the file:

   ```
   username=weblogic_iam
   password=password
   ```

5. Save the file.

Adding a Load Balancer Certificate to JDK Trust Stores for OIG

Some OIG Products require that the SSL certificate used by the load balancer be added to the trusted certificates in the JDK. To add the certificate, do the following:

1. Create a directory to hold user created keystores and certificates.

   For example:

   ```
   mkdir SHARED_CONFIG_DIR/keystores
   ```

2. Obtain the certificate from the load balancer. You can obtain the load balancer certificate from using a browser, such as Firefox. However, the easiest way to obtain the certificate is to use the `openssl` command. The syntax of the command is as follows:

   ```
   openssl s_client -connect  LOADBALANCER
         -showcerts </dev/null 2>/dev/null|openssl x509  -outform PEM >
         SHARED_CONFIG_DIR/keystores/LOADBALANCER.pem
   ```

For example:

```
openssl s_client -connect
      prov.example.com:443 -showcerts </dev/null 2>/dev/null|openssl  x509
-outform PEM >
      SHARED_CONFIG_DIR/keystores/prov.example.com.pem
```

This command saves the certificate to a file called `prov.example.com.pem` in `SHARED_CONFIG_DIR/keystores`.

3. Load the certificate into the JDK and Node Manager Trust Stores by running the following command to import the CA certificate file, `login.example.com.pem`, into the JAVA_HOME.
4. Set **JAVA_HOME** to *JAVA_HOME*.
5. Set **PATH** to include `JAVA_HOME/bin`.
6. Execute the following command to import the certificate into the Java trust store.

```
keytool -importcert -alias
      prov.example.com -file SHARED_CONFIG_DIR/keystores/
prov.example.com.pem  -trustcacerts
      -keystore $JAVA_HOME/jre/lib/security/cacerts
```

7. Enter a password for the keystore. The default password for the JDK is *changeit*. The default password for the Node Manager keystores is *COMMON_IAM_PASSWORD*. You will be prompted to confirm that the certificate is valid.

Configuring the WebLogic Proxy Plug-In

Before you can validate that requests are routed correctly through the Oracle HTTP Server instances, you must set the `WebLogic Plug-In Enabled` parameter. It is recommended to set the `WebLogic Plug-In Enabled` parameter at the domain level. Any clusters or servers not using the plugin via the web-tier can have their `WebLogic Plug-In Enabled` parameter value set to `no` on an exception basis as needed.

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Domain Structure** pane, click on the top-level domain node.
3. Click **Lock & Edit** in the Change Center.
4. Click on the Domain Name.
5. Click on the **Web Applications** tab.
6. Locate and select the **WebLogic PlugIn Enabled** option.
7. Click **Save**.
8. Click **Activate Changes** in the Change Center.
9. Restart the Administration Server.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

Verification of Manual Failover of the Administration Server

After you configure the domain, test failover by following the steps that are described in Verifying Manual Failover of the Administration Server.

15 Configuring Oracle HTTP Server for an Enterprise Deployment

For an enterprise deployment, Oracle HTTP Server must be installed on each of the web tier hosts and configured as Oracle HTTP standalone domains on each host.

The Oracle HTTP Server instances on the web tier direct HTTP requests from the hardware load balancer to specific Managed Servers in the application tier.

Before you configure Oracle HTTP Server, be sure to review About Web Tier.

> **Note:**
>
> As of Fusion Middleware 12.2.1.4.0, Oracle Traffic Director has been deprecated. For an enterprise deployment, use Oracle HTTP Server.

- Variables Used When Configuring the Oracle HTTP Server
  You reference these directory variables as you perform the different tasks explained in this chapter.
- About the Oracle HTTP Server Domains
  In an enterprise deployment, each Oracle HTTP Server instance is configured on a separate host and in its own standalone domain. This allows for a simple configuration that requires a minimum amount of configuration and a minimum amount of resources to run and maintain.
- Installing a Supported JDK
- Installing Oracle HTTP Server on WEBHOST1
  Install the Oracle HTTP Server software on the web tier by using the Oracle Universal Installer. Verify the installation after you complete the procedure.
- Creating an Oracle HTTP Server Domain on WEBHOST1
  You can create a new Oracle HTTP Server standalone domain on the first web tier host by using the Configuration Wizard.
- Installing and Configuring an Oracle HTTP Server Domain on WEBHOST2
  After you install Oracle HTTP Server and configure a domain on WEBHOST1, then you must also perform the same tasks on WEBHOST2.
- Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2
  It is important to understand how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.
- Backing Up the Configuration
  It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

- Configuring Oracle HTTP Server to Route Requests to the Application Tier
  Update the Oracle HTTP Server configuration files so that the web server instances route requests to the servers in the domain.
- Configuring Oracle HTTP Server for Oracle Access Manager Managed Servers
  You have to configure Oracle HTTP Server for the Oracle Access Manager Managed Servers to ensure they route requests correctly to the Oracle Access Management cluster.
- Configuring Oracle HTTP Server for Oracle Identity Governance Managed Servers
- Validating the Virtual Server Configuration and Access to the Consoles
  Validate the virtual server configuration on the load balancer, and the access to the management console and the Administration Server.
- Restarting the OHS Instances on OHSHOST1 and OHSHOST2
  Restart the Oracle HTTP Server (OHS) instances on both OHSHOST1 and OHSHOST2.
- Sample Virtual Host Files

Variables Used When Configuring the Oracle HTTP Server

You reference these directory variables as you perform the different tasks explained in this chapter.

The values for several directory variables are defined in File System and Directory Variables Used in This Guide.

- *WEB_ORACLE_HOME*
- *WEB_DOMAIN_HOME*
- *JAVA _HOME*

In addition, you reference the following virtual IP (VIP) address and host names:

- ADMINVHN
- WEBHOST1
- WEBHOST2

About the Oracle HTTP Server Domains

In an enterprise deployment, each Oracle HTTP Server instance is configured on a separate host and in its own standalone domain. This allows for a simple configuration that requires a minimum amount of configuration and a minimum amount of resources to run and maintain.

> **Note:**
>
> Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system and JAVA_HOME is set on the web tier hosts.

For more information about the role and configuration of the Oracle HTTP Server instances in the web tier, see Understanding the Web Tier.

Installing a Supported JDK

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system.

- Locating and Downloading the JDK Software
- Installing the JDK Software
  Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

Locating and Downloading the JDK Software

To find a certified JDK, see the certification document for your release on the Oracle Fusion Middleware Supported System Configurations page.

After you identify the Oracle JDK for the current Oracle Fusion Middleware release, you can download an Oracle JDK from the following location on Oracle Technology Network:

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the following locations:

On the local storage device for each of the Web tier host computers. The Web tier host computers, which reside in the DMZ, do not necessarily have access to the shared storage on the application tier.

See the Understanding the Recommended Directory Structure for an Enterprise Deployment.

To install JDK 1.8.0_211:

1. Change directory to the location where you downloaded the JDK archive file.

   ```
   cd download_dir
   ```

2. Unpack the archive into the JDK home directory, and then run the following commands:

   ```
   tar -xzvf jdk-8u201-linux-x64.tar.gz
   ```

   Note that the JDK version listed here was accurate at the time this document was published. For the latest supported JDK, see the *Oracle Fusion Middleware System Requirements and Specifications* for the current Oracle Fusion Middleware release.

3. Move the JDK directory to the recommended location in the directory structure.

   For example:

   ```
   mv ./jdk1.8.0_211 /u02/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

For example:

```
export JAVA_HOME=/u02/oracle/products/jdk
export PATH=$JAVA_HOME/bin:$PATH
```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

   ```
   java -version
   ```

   The Java version in the output should be displayed as `1.8.0_211`.

Installing Oracle HTTP Server on WEBHOST1

Install the Oracle HTTP Server software on the web tier by using the Oracle Universal Installer. Verify the installation after you complete the procedure.

- Starting the Installer on WEBHOST1
- Navigating the Oracle HTTP Server Installation Screens
- Verifying the Oracle HTTP Server Installation

Starting the Installer on WEBHOST1

To start the installation program, perform the following steps.

1. Log in to WEBHOST1.
2. Go to the directory in which you downloaded the installation program.
3. Enter the following command to launch the installation program:

   ```
   ./fmw_12.2.1.3.0_ohs_linux64.bin
   ```

   When the installation program appears, you are ready to begin the installation.

Navigating the Oracle HTTP Server Installation Screens

The following table lists the screens in the order that the installation program displays them.

If you need additional help with any of the installation screens, click the Help button on the screen.

Table 15-1 Oracle HTTP Server Installation Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears if you install any Oracle product on this host for the first time. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location. See Understanding the Oracle Central Inventory in *Installing Software with the Oracle Universal Installer*. **Note:** Oracle recommends that you configure the central inventory directory within the products directory. Example: `/u02/oracle/products/oraInventory` You may also need to execute the `createCentralinventory.sh` script as root from the `oraInventory` folder after the installer completes.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search the local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For the purposes of an enterprise deployment, enter the value of the *WEB_ORACLE_HOME* variable listed in Table 8-3.
Installation Type	Select **Standalone HTTP Server (Managed independently of WebLogic server)**. This installation type allows you to configure the Oracle HTTP Server instances independently from any other existing Oracle WebLogic Server domains.
JDK Selection	For the value of JDK Home, enter the value of *JAVA_HOME* that you set when installing the JDK software.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, verify that your host computers and the required software meet the system requirements and certification information described in Host Computer Hardware Requirements and Operating System Requirements for the Enterprise Deployment Topology.

Table 15-1 (Cont.) Oracle HTTP Server Installation Screens

Screen	Description
Installation Summary	Use this screen to verify the installation options that you selected. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file. Response files can be used later in a silent installation situation. See Using the Oracle Universal Installer in Silent Mode in *Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click **Finish** to close the installer.

Verifying the Oracle HTTP Server Installation

Verify that the Oracle HTTP Server installation completed successfully by validating the `WEB_ORACLE_HOME` folder contents.

Run the following command to compare the installed folder structure with the following list:

```
ls --format=single-column WEB_ORACLE_HOME
```

The following files and directories are listed in theOracle HTTP Server Oracle Home:

```
bin
cdata
cfgtoollogs
crs
css
cv
has
install
inventory
jlib
ldap
lib
network
nls
ohs
OPatch
oracle_common
oracore
oraInst.loc
oui
perl
plsql
plugins
precomp
QOpatch
racg
rdbms
slax
```

```
sqlplus
srvm
webgate
wlserver
xdk
```

Creating an Oracle HTTP Server Domain on WEBHOST1

You can create a new Oracle HTTP Server standalone domain on the first web tier host by using the Configuration Wizard.

- Starting the Configuration Wizard on WEBHOST1
- Navigating the Configuration Wizard Screens for an Oracle HTTP Server Domain

Starting the Configuration Wizard on WEBHOST1

To start the Configuration Wizard, navigate to the following directory and start the WebLogic Server Configuration Wizard, as follows:

```
cd WEB_ORACLE_HOME/oracle_common/common/bin
./config.sh
```

Navigating the Configuration Wizard Screens for an Oracle HTTP Server Domain

Oracle recommends that you create a standalone domain for the Oracle HTTP Server instances on each web tier host.

The following topics describe how to create a new standalone Oracle HTTP Server domain:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Templates
- Task 3, Selecting the JDK for the Web Tier Domain.
- Task 4, Configuring System Components
- Task 5, Configuring OHS Server
- Task 7, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 8, Writing Down Your Domain Home

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a new domain**.
In the **Domain Location** field, enter the value assigned to the *WEB_DOMAIN_HOME* variable.
Note the following:

- The Configuration Wizard creates the new directory that you specify here.
- Create the directory on local storage, so the web servers do not have any dependencies on storage devices outside the DMZ.

Task 2 Selecting the Configuration Templates
On the Templates screen, select **Oracle HTTP Server (Standalone) - 12.2.1.4.0 [ohs]**.

Tip:

More information about the options on this screen can be found in Templates in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Selecting the JDK for the Web Tier Domain.
Select the Oracle HotSpot JDK installed in the `/u02/oracle/products/jdk` directory prior to the Oracle HTTP Server installation.

Task 4 Configuring System Components
On the System Components screen, configure one Oracle HTTP Server instance. The screen should, by default, have a single instance defined. This is the only instance that you need to create.

1. The default instance name in the **System Component** field is `ohs1`. Use this default name when you configure `WEBHOST1`.
2. Make sure that `OHS` is selected in the **Component Type** field.
3. If an application is not responding, use the **Restart Interval Seconds** field to specify the number of seconds to wait before you attempt a restart if an application is not responding.
4. Use the **Restart Delay Seconds** field to specify the number of seconds to wait between restart attempts.

Task 5 Configuring OHS Server
Use the OHS Server screen to configure the OHS servers in your domain:

1. Select **ohs1** from the **System Component** drop-down menu.
2. In the **Listen Address** field, enter *`WEBHOST1`*.

   All the remaining fields are prepopulated, but you can change the values as required for your organization. See OHS Server in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.
3. In the **Server Name** field, verify the value of the listen address and listen port.

   It should appear as follows:

   ```
   http://WEBHOST1:7777
   ```

Task 6 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type, and specify the user name and password for the Node Manager.

Note:

For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.
For information about Node Manager configuration, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Task 7 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains detailed configuration information for the domain that you are about to create. Review the details of each item on the screen and verify that the information is correct.
If you need to make any changes, you can go back to any previous screen either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation does not begin until you click **Create**.
In the Configuration Progress screen, click **Next** when it finishes.

> **Tip:**
>
> More information about the options on this screen can be found in Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Writing Down Your Domain Home
The Configuration Success screen shows the domain home location.
Make a note of the information provided here, as you need it to start the servers and access the Administration Server.
Click **Finish** to close the Configuration Wizard.

Installing and Configuring an Oracle HTTP Server Domain on WEBHOST2

After you install Oracle HTTP Server and configure a domain on WEBHOST1, then you must also perform the same tasks on WEBHOST2.

1. Log in to WEBHOST2 and install Oracle HTTP Server by using the instructions in Installing Oracle HTTP Server on WEBHOST1.
2. Configure a new standalone domain on WEBHOST2 by using the instructions in Creating a Web Tier Domain on WEBHOST1.

   Use the name `ohs2` for the instance on WEBHOST2, and be sure to replace all occurrences of WEBHOST1 with WEBHOST2 and all occurrences of `ohs1` with `ohs2` in each of the examples.

Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2

It is important to understand how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.

- Starting the Node Manager on WEBHOST1 and WEBHOST2
- Starting the Oracle HTTP Server Instances

Starting the Node Manager on WEBHOST1 and WEBHOST2

Before you can start the Oracle HTTP Server instances, you must start the Node Manager on WEBHOST1 and WEBHOST2:

1. Log in to WEBHOST1 and navigate to the following directory:

```
WEB_DOMAIN_HOME/bin
```

2. Start the Node Manager as shown in the following sections by using `nohup` and `nodemanager.out` as an example output file:

   ```
   nohup WEB_DOMAIN_HOME/bin/startNodeManager.sh > WEB_DOMAIN_HOME/nodemanager/
   nodemanager.out 2>&1 &
   ```

3. Log in to WEBHOST2 and perform steps 1 and 2.

See Advanced Node Manager Configuration in *Administering Node Manager for Oracle WebLogic Server*.

Starting the Oracle HTTP Server Instances

To start the Oracle HTTP Server instances:

1. Navigate to the following directory on WEBHOST1:

   ```
   WEB_DOMAIN_HOME/bin
   ```

   For more information about the location of the WEB_DOMAIN_HOME directory, see File System and Directory Variables Used in This Guide.

2. Enter the following command:

   ```
   ./startComponent.sh ohs1
   ```

   Note:

   Every time you start the Oracle HTTP server, you will be asked for the Node Manager password. If you do not wish this behaviour, then use the following command the first time you start the Oracle HTTP server:

   ```
   ./startComponent.sh ohs1 storeUserConfig
   ```

   This time when you enter the Node Manager password, it will be encrypted and stored. Future start and stop of the Oracle HTTP server will not require you to enter the Node Manager password.

   Note:

   For more information, see Storing Your Node Manager Password.

3. When prompted, enter the Node Manager password.

4. Repeat steps 1 through 3 to start the `ohs2` instance on WEBHOST2. See Starting Oracle HTTP Server Instances in *Administering Oracle HTTP Server*.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

Configuring Oracle HTTP Server to Route Requests to the Application Tier

Update the Oracle HTTP Server configuration files so that the web server instances route requests to the servers in the domain.

- About the Oracle HTTP Server Configuration for an Enterprise Deployment
- Modifying the httpd.conf File to Include Virtual Host Configuration Files
- Modifying the httpd.conf File to Set Server Runtime Parameters
- Creating the Virtual Host Configuration Files
- Configuring Routing to the Administration Server and Oracle Web Services Manager

About the Oracle HTTP Server Configuration for an Enterprise Deployment

The following topics provide overview information about the changes that are required to the Oracle HTTP Server configuration files in an enterprise deployment.

- Purpose of the Oracle HTTP Server Virtual Hosts
- About the WebLogicCluster Parameter of the <VirtualHost> Directive
- Recommended Structure of the Oracle HTTP Server Configuration Files

Purpose of the Oracle HTTP Server Virtual Hosts

The reference topologies in this guide require that you define a set of virtual servers on the hardware load balancer. You can then configure Oracle HTTP Server to recognize requests to specific virtual hosts (that map to the load balancer virtual servers) by adding `<VirtualHost>` directives to the Oracle HTTP Server instance configuration files.

For each Oracle HTTP Server virtual host, you define a set of specific URLs (or context strings) that route requests from the load balancer through the Oracle HTTP Server instances to the appropriate Administration Server or Managed Server in the Oracle WebLogic Server domain.

About the WebLogicCluster Parameter of the <VirtualHost> Directive

A key parameter of the Oracle HTTP Server `<VirtualHost>` directive is the `WebLogicCluster` parameter, which is part of the WebLogic Proxy Plug-In for Oracle HTTP Server. When you configure Oracle HTTP Server for an enterprise deployment, consider the following information when you add this parameter to the Oracle HTTP Server configuration files.

The servers specified in the `WebLogicCluster` parameter are important only at startup time for the plug-in. The list needs to provide at least one running cluster member for the plug-in to discover other members of the cluster. When you start the Oracke HTTP server, the listed cluster member must be running.. Oracle WebLogic Server and the plug-in work together to update the server list automatically with new, failed, and recovered cluster members.

Some example scenarios:

- Example 1: If you have a two-node cluster and then add a third member, you do not need to update the configuration to add the third member. The third member is discovered on the fly at runtime.
- Example 2: You have a three-node cluster but only two nodes are listed in the configuration. However, if both listed nodes are down when you start Oracle HTTP Server, then the plug-in would fail to route to the cluster. You must ensure that at least one of the listed nodes is running when you start Oracle HTTP Server.

  If you list all members of the cluster, then you guarantee you can route to the cluster, assuming at least one member is running when Oracle HTTP Server is started.

Recommended Structure of the Oracle HTTP Server Configuration Files

Rather than adding multiple virtual host definitions to the `httpd.conf` file, Oracle recommends that you create separate, smaller, and more specific configuration files for each of the virtual servers required for the products that you are deploying. This avoids populating an already large `httpd.conf` file with additional content, and it can make troubleshooting configuration problems easier.

For example, in a typical Oracle Fusion Middleware Infrastructure domain, you can add a specific configuration file called `admin_vh.conf` that contains the virtual host definition for the Administration Server virtual host (ADMINVHN).

Modifying the httpd.conf File to Include Virtual Host Configuration Files

Perform the following tasks to prepare the `httpd.conf` file for the additional virtual hosts required for an enterprise topology:

1. Log in to WEBHOST1.
2. Locate the `httpd.conf` file for the first Oracle HTTP Server instance (`ohs1`) in the domain directory:

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
   ```

3. Verify if the `httpd.conf` file has the appropriate configuration as follows:

   a. Run the following command to verify the `ServerName` parameter, be sure that it is set correctly, substituting the correct value for the current WEBHOST*n*:

   ```
   grep "ServerName http" httpd.conf
   ServerName http://WEBHOST1:7777
   ```

   b. Run the following command to verify there is an include statement that includes all `*.conf` files from the moduleconf subdirectory:

   ```
   grep moduleconf httpd.conf
   IncludeOptional "moduleconf/*.conf"
   ```

c. If either validation fails to return results, or returns results that are commented out, open the httpd.conf file in a text editor and make the required changes in the appropriate locations.

```
#
# ServerName gives the name and port that the server uses to identify
itself.
# This can often be determined automatically, but we recommend you
specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address
here.
#
ServerName http://WEBHOST1:7777
#  and at the end of the file:
# Include the admin virtual host (Proxy Virtual Host) related
configuration
include "admin.conf"
IncludeOptional "moduleconf/*.conf"
```

d. Save the httpd.conf file.

4. Log in to WEBHOST2 and perform steps 2 and 3 for the httpd.conf file, replacing any occurrences of WEBHOST1 or ohs1 with WEBHOST2 or ohs2 in the instructions as necessary.

Modifying the httpd.conf File to Set Server Runtime Parameters

Out of the box, the Oracle HTTP Server comes configured with a number of values which effect how the server behaves when it is running. For most of the deployments, these values are sufficient. However, in an Oracle Identity and Access Management deployment, it is recommended that you update these values by doing the following:

1. Log in to WEBHOST1.
2. Locate the httpd.conf file for the first Oracle HTTP Server instance (ohs1) in the domain directory:

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
   ```

3. Locate the section of the file with the following line:

   ```
   <IfModule mpm_worker_module>
   ```

4. Update the entries in this section to reflect the following:

   ```
   <IfModule mpm_worker_module>
     ServerLimit            20
     StartServers           10
     MaxClients             1500
     MinSpareThreads        200
     MaxSpareThreads        800
     ThreadsPerChild        250
     ThreadLimit            250
     MaxRequestsPerChild    1000
     MaxRequestWorkers      400
     MaxConnectionsPerChild  0
    </IfModule>
   ```

5. Update the following values:
   - `MaxKeepAliveRequests 0`
   - `Timeout 300`
   - `KeepAliveTimeout 10`
6. Save the `httpd.conf` file.
7. Log in to `WEBHOST2` and perform steps 2 and 3 for the `httpd.conf` file, replacing any occurrences of `WEBHOST1` or `ohs1` with `WEBHOST2` or `ohs2` in the instructions as necessary.

Creating the Virtual Host Configuration Files

To create the virtual host configuration files:

> **Note:**
>
> Before you create the virtual host configuration files, be sure that you have configured the virtual servers on the load balancer, as described in Purpose of the Oracle HTTP Server Virtual Hosts.

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (`ohs1`):

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
   ```

2. If you are configuring Oracle Access Management, create the `iadadmin_vh.conf` file and add the following directive:

   ```
   <VirtualHost WEBHOST1.example.com:7777>
       ServerName http://iadadmin.example.com:80
       ServerAdmin you@your.address
       RewriteEngine On
       RewriteOptions inherit
       UseCanonicalName On
   </VirtualHost>
   ```

3. If you are configuring Oracle Access Management, create the `login_vh.conf` file and add the following directive:

   ```
   <VirtualHost WEBHOST1.example.com:7777>
       ServerName https://login.example.com:443
       ServerAdmin you@your.address
       RewriteEngine On
       RewriteOptions inherit
       UseCanonicalName On
   </VirtualHost>
   ```

4. If you are configuring Oracle Identity Governance, create the `igdadmin_vh.conf` file, and add the following directive:

   ```
   <VirtualHost WEBHOST1.example.com:7777>
       ServerName http://igdadmin.example.com:80
       ServerAdmin you@your.address
       RewriteEngine On
       RewriteOptions inherit
   ```

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

5. If you are configuring Oracle Identity Governance, create the `prov_vh.conf` file, and add the following directive:

```
<VirtualHost WEBHOST1.example.com:7777>
    ServerName https://prov.example.com:443
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On
</VirtualHost>
```

6. If you are configuring Oracle Identity Governance, create the `igdinternal_vh.conf` file, and add the following directive:

```
<VirtualHost WEBHOST1.example.com:7777>
    ServerName http://igdinternal.example.com:7777
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
</VirtualHost>
```

Configuring Routing to the Administration Server and Oracle Web Services Manager

To enable Oracle HTTP Server to route to the Administration Server and the WSM-PM_Cluster, which contain the WLS_WSM managed servers, you must add a set of `<Location>` directives and add the `WebLogicCluster` parameter to the list of nodes in the cluster.

To set the `WebLogicCluster` parameter:

1. Log in to WEBHOST1, and change directory to the following location:

   cd *WEB_DOMAIN_HOME*/config/fmwconfig/components/OHS/ohs1/moduleconf/

2. Add the following directives to the `iadadmin_vh.conf` file within the `<VirtualHost>` tags:

```
# Admin Server and EM
<Location /console>
    WLSRequest ON
    WebLogicHost IADADMINVHN.example.com
    WeblogicPort 7001
</Location>

<Location /consolehelp>
    WLSRequest ON
    WebLogicHost IADADMINVHN.example.com
    WeblogicPort 7001
</Location>

<Location /em>
    WLSRequest ON
    WebLogicHost IADADMINVHN.example.com
    WeblogicPort 7001
</Location>
```

3. Add the following directives to the `igdadmin_vh.conf` file within the `<VirtualHost>` tags:

   ```
   # Admin Server and EM
   <Location /console>
       WLSRequest ON
       WebLogicHost IGDADMINVHN.example.com
       WeblogicPort 7101
   </Location>

   <Location /consolehelp>
       WLSRequest ON
       WebLogicHost IGDADMINVHN.example.com
       WeblogicPort 7101
   </Location>

   <Location /em>
       WLSRequest ON
       WebLogicHost IGDADMINVHN.example.com
       WeblogicPort 7101
   </Location>
   ```

4. Add the following directives to the `igdinternal_vh.conf` file within the `<VirtualHost>` tag:

   > **Note:**
   >
   > Configure the port numbers appropriately, as assigned for your static or dynamic cluster. Dynamic clusters with the Calculate Listen Port option selected have incremental port numbers for each dynamic managed server that is created automatcially.
   >
   > The WebLogicCluster directive needs only a sufficient number of redundant server:port combinations to guarantee initial contact in case of a partial outage. The actual total list of cluster members is retrieve automatically upon first contact with any given node.

   ```
   # WSM-PM
   <Location /wsm-pm>
       WLSRequest ON
       WebLogicCluster OIMHOST1.example.com:7010,OIMHOST2.example.com:7010
       WLProxySSL OFF
       WLProxySSLPassThrough OFF
   </Location>
   ```

   For more information about the WebLogicCluster parameter in this example, see About the WebLogicCluster Parameter of the <VirtualHost> Directive.

5. Copy the files iadadmin_vh.conf, igdadmin_vh.conf, and igdinternal_vh.conf edited in step 2, 3, and 4, respectively, to the configuration directory for the second Oracle HTTP Server instance (ohs2) on WEBHOST2:

   ```
   WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf/
   ```

6. Edit each of the files copied in the previous step on WEBHOST2 and change the `<VirtualHost>` directive references from `WEBHOST1.example.com:7777` to `WEBHOST2.example.com:7777`.

Configuring Oracle HTTP Server for Oracle Access Manager Managed Servers

You have to configure Oracle HTTP Server for the Oracle Access Manager Managed Servers to ensure they route requests correctly to the Oracle Access Management cluster.

- Configuring Oracle HTTP Server for the WLS_OAM Managed Servers
- Validating Access Through the Load Balancer

Configuring Oracle HTTP Server for the WLS_OAM Managed Servers

To configure the Oracle HTTP Server instances in the web tier so they route requests correctly to the Oracle Access Management cluster, use the following procedure to create an additional Oracle HTTP Server configuration file that creates and defines the parameters of the `login.example.com` virtual server. To configure Oracle HTTP Server for the WLS_OAM Managed Servers:

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (`ohs1`).

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
   ```

   > **Note:**
   >
   > There are separate directories for configuration and runtime instance files. The runtime files under the `.../OHS/instances/ohsn/*` folder should not be edited directly. Edit only the `.../OHS/ohsn/*` configuration files.

2. In the `login_vh.conf` file, add the following lines between the `<VirtualHost>` and `</VirtualHost>` tags:

   ```
   #OAM Entries
   <Location /oam>
     WLSRequest ON
     WLProxySSL ON
     WLProxySSLPassThrough ON
     WLCookieName OAMJSESSIONID
     WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
   </Location>

   <Location /oamfed>
     WLSRequest ON
     WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
     WLCookieName OAMJSESSIONID
     WLProxySSL ON
     WLProxySSLPassThrough ON
   </Location>

   # OAM Forgotten Password Page
   <Location /otpfp/>
     WLSRequest ON
   ```

```
  WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
  WLCookieName OAMJSESSIONID
  WLProxySSL ON
  WLProxySSLPassThrough ON
</Location>

<Location /ms_oauth>
  WLSRequest ON
  WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
  WLCookieName OAMJSESSIONID
  WLProxySSL ON
  WLProxySSLPassThrough ON
</Location>
<Location /iam>
  WLSRequest ON
  WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
  WLCookieName OAMJSESSIONID
  WLProxySSL ON
  WLProxySSLPassThrough ON
</Location>
```

3. In the `iadadmin_vh.conf` file, add the following lines between the `<VirtualHost>` and `</VirtualHost>` tags:

```
<Location /oamconsole>
  WLSRequest ON
  WebLogicHost IADADMINVHN.example.com
  WeblogicPort 7001
</Location>

<Location /access>
  WLSRequest ON
  WebLogicCluster OAMHOST1.example.com:14150,OAMHOST2.example.com:14150
  WLCookieName OAMJSESSIONID
</Location>

# Required for Multi-Datacenter
<Location /oam/services>
  WLSRequest ON
  WebLogicHost IADADMINVHN.example.com
  WeblogicPort 7001
</Location>
```

> **Note:**
>
> Location `/oam/services` is required only for mulit-datacenter deployments.

4. Copy the `iadadmin_vh.conf` file and `login_vh.conf` to the configuration directory for the second Oracle HTTP Server instance (`ohs2`):

   WEB_DOMAIN_HOME/config/fmwconfig/components/**ohs2**/moduleconf/

5. Edit the `login_vh.conf` and `iadadmin_vh.conf` change any references of WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.

6. Restart the Oracle HTTP server instances on WEBHOST1 and WEBHOST2.

Validating Access Through the Load Balancer

You should verify URLs to ensure that appropriate routing and failover is working from Oracle HTTP Server to OAM_Cluster.

- Verifying the URLs

Verifying the URLs

To verify the URLs:

1. While WLS_OAM2 is running, stop WLS_OAM1 using the WebLogic Server Administration Console.
2. Access `https://login.example.com/oam/server/logout`.
3. Start WLS_OAM1 from the WebLogic Server Administration Console.
4. Stop WLS_OAM2 from the WebLogic Server Administration Console.
5. Access `http://login.example.com/oam/server/logout`.

You can verify the cluster node to which you were directed after the traffic balancing provided through your load balancer and then again through the web tier.

Configuring Oracle HTTP Server for Oracle Identity Governance Managed Servers

To configure the Oracle HTTP Server instances in the Web tier so they route requests correctly to the Oracle SOA Suite cluster, use the following procedure to create an additional Oracle HTTP Server configuration file that creates and defines the parameters of the `https://igdinternal.example.com:7777` virtual server.

This procedure assumes you performed the Oracle HTTP Server configuration tasks described in Configuring Oracle HTTP Server to Route Requests to the Application Tier.

To create the virtual host configuration file so requests are routed properly to the Oracle Identity Governance clusters:

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (OHS_1):

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
   ```

2. Edit the file `prov_vh.conf` and add the following directives inside the `<VirtualHost>` tags:

Note:

- The URL entry for `/workflow` is optional. It is for workflow tasks associated with Oracle ADF task forms. The `/workflow` URL itself can be a different value, depending on the form.
- Configure the port numbers appropriately, as assigned for your static or dynamic cluster. Dynamic clusters with the Calculate Listen Port option selected will have incremental port numbers for each dynamic managed server that you create.

  The WebLogicCluster directive needs only a sufficient number of redundant *server:port* combinations to guarantee an initial contact in case of a partial outage. The actual total list of cluster members is retrieved automatically on the first contact with any given node.

```
<Location /identity>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.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>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /HTTPClnt>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

# Requests webservice URL
<Location /reqsvc>
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLProxySSL ON
    WLProxySSLPassThrough ON
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /FacadeWebApp>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /iam>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>

<Location /OIGUI>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
</Location>
```

The `prov_vh.conf` file will appear as it does in Step 2.

3. In the `igdadmin_vh.conf` file, add the following lines between `<VirtualHost>` and `</VirtualHost>` tags:

```
## Entries Required by Oracle Identity Governance
<Location /oim>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /iam>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /sysadmin>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /admin>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster oimhost1.example.com:14000,oimhost2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

# xlWebApp - Legacy 9.x webapp (struts based)
<Location /xlWebApp>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>
```

```
# OIM self service console
<Location /identity>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /OIGUI>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

# Nexaweb WebApp - used for workflow designer and DM
<Location /Nexaweb>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /FacadeWebApp>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster oimhost1.example.com:14000,oimhost2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

# Scheduler webservice URL
<Location /SchedulerService-web>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>
```

4. In the `igdinternal_vh.conf` file, add the following lines between the `<VirtualHost>` and `</VirtualHost>` tags:

```
## Entries Required by Oracle Identity Governance
#SOA Callback webservice for SOD - Provide the SOA Managed Server Ports

<Location /sodcheck>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
</Location>

# OIM, role-sod profile
<Location /role-sod>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

# Callback webservice for SOA. SOA calls this when a request is approved/rejected
# Provide the SOA Managed Server Port
```

```
<Location /workflowservice>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
</Location>

# used for FA Callback service.
<Location /callbackResponseService>
    WLSRequest ON
    WLCookieName    oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

# spml xsd profile
<Location /spml-xsd>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

# OIM, spml dsml profile
<Location /spmlws>
    WLSRequest ON
    PathTrim /weblogic
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /reqsvc>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
</Location>

# SOA Infra
<Location /soa-infra>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/OHS/component/oim_component.log"
</Location>

# UMS Email Support
<Location /ucs>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/OHS/component/oim_component.log"
</Location>

<Location /provisioning-callback>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>
```

```
<Location /CertificationCallbackService>
   WLSRequest ON
   WLCookieName oimjsessionid
   WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
   WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /IdentityAuditCallbackService>
   WLSRequest ON
   WLCookieName oimjsessionid
   WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
   WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

# SOA Callback webservice for SOD - Provide the SOA Managed Server Ports
  <Location /soa/composer>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
  </Location>

  <Location /integration>
    SetHandler weblogic-handler
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLCookieName oimjsessionid
  </Location>

  <Location /sdpmessaging/userprefs-ui>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
  </Location>

<Location /iam>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>

<Location /ws_utc>
SetHandler weblogic-handler
  WLCookieName oimjsessionid
  WebLogicCluster OIMHOST1:8001,OIMHOST2:8001
  WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
</Location>
```

5. Copy the `igdadmin_vh.conf`, `igdinternal_vh.conf`, and `prov_vh.conf` files to the configuration directory for the second Oracle HTTP Server instance (ohs2):

   WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/**ohs2**/moduleconf/

6. Edit the `igdadmin_vh.conf`, `prov_vh.conf`, and `igdinternal_vh.conf` files and change any references to WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.

7. Restart the Oracle HTTP servers on WEBHOST1 and WEBHOST2.

> **Note:**
>
> If internal invocations are going to be used in the system, add the appropriate locations to the sointernal virtual host.

Validating the Virtual Server Configuration and Access to the Consoles

Validate the virtual server configuration on the load balancer, and the access to the management console and the Administration Server.

From the load balancer, access the following URLs to ensure that your load balancer and Oracle HTTP Server are configured properly. These URLs should show the initial Oracle HTTP Server 12*c* web page.

- `https://login.example.com/index.html`
- `https://prov.example.com/index.html`
- `http://iadadmin.example.com/index.html`
- `http://igdadmin.example.com/index.html`

Use the following URLs to the hardware load balancer to display the Oracle WebLogic Server Administration Console, and log in using the Oracle WebLogic Server `iadadmin` credentials:

- `http://iadadmin.example.com/console`
- `http://iadadmin.example.com/em`

This validates that the `iadadmin.example.com` virtual host on the load balancer is able to route requests to the Oracle HTTP Server instances on the web tier, which in turn can route requests for the Oracle WebLogic Server Administration Console to the Administration Server in the application tier.

Similarly, you should be able to access the WebLogic Server Administration Console and Fusion Middleware Control for the `igdadmin` virtual host using the following URLs:

- `http://igdadmin.example.com/console`
- `http://igdadmin.example.com/em`

Restarting the OHS Instances on OHSHOST1 and OHSHOST2

Restart the Oracle HTTP Server (OHS) instances on both OHSHOST1 and OHSHOST2.

To do this:

1. Restart the ohs1 instance by doing the following:
   a. Change directory to the following location:

      ```
      cd WEB_DOMAIN_HOME/bin
      ```

b. Enter the following commands to stop and start the instance:

```
./stopComponent.sh ohs1
```

```
./startComponent.sh ohs1
```

2. Restart the ohs2 instance by doing the following:

   a. Change directory to the following location:

```
cd WEB_DOMAIN_HOME/bin
```

   b. Enter the following commands to stop and start the instance:

```
./stopComponent.sh ohs2
```

```
./startComponent.sh ohs2
```

Sample Virtual Host Files

This section lists the examples used in

Example 1 iadadmin_vh.conf file

```
<VirtualHost WEBHOST1.example.com:7777>
    ServerName iadadmin.example.com:80
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On

# Admin Server and EM
    <Location /console>
    WLSRequest ON
    WebLogicHost IADADMINVHN.example.com in example1
    WeblogicPort 7001
    </Location>

    <Location /consolehelp>
    WLSRequest ON
    WebLogicHost.example.com IADADMINVHN
    WeblogicPort 7001
    </Location>

    <Location /em>
    WLSRequest ON
    WebLogicHost.example.com IADADMINVHN
    WeblogicPort 7001
    </Location>

    <Location /oamconsole>
    WLSRequest ON
    WebLogicHost IADADMINVHN.example.com
    WeblogicPort 7001
    </Location>
    <Location /access>
    WLSRequest ON
    WebLogicCluster OAMHOST1.example.com:14150,OAMHOST2.example.com:14150
    WLCookieName OAMJSESSIONID
    </Location>

# Required for Multi-Datacenter
    <Location /oam/services>
    WLSRequest ON
    WebLogicHost IADADMINVHN.example.com
    WeblogicPort 7001
    </Location>
</VirtualHost>
```

Example 2 `igdadmin_vh.conf file`

```
VirtualHost WEBHOST1.example.com:7777>
    ServerName igdadmin.example.com:80
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit
    UseCanonicalName On

# Admin Server and EM
    <Location /console>
    WLSRequest ON
    WebLogicHost IGDADMINVHN.example.com
    WeblogicPort 7101
    </Location>

    <Location /consolehelp>
    WLSRequest ON
    WebLogicHost IGDADMINVHN.example.com
    WeblogicPort 7101
    </Location>

    <Location /em>
    WLSRequest ON
    WebLogicHost IGDADMINVHN.example.com
    WeblogicPort 7101
    </Location>

    <Location /oim>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

    <Location /iam>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

    <Location /sysadmin>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

    <Location /admin>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster oimhost1.example.com:14000,oimhost2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

# xlWebApp - Legacy 9.x webapp (struts based)
    <Location /xlWebApp>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

# OIM self service console
    <Location /identity>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

    <Location /OIGUI>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

# Nexaweb WebApp - used for workflow designer and DM
    <Location /Nexaweb>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

    <Location /FacadeWebApp>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster oimhost1.example.com:14000,oimhost2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

# Scheduler webservice URL
    <Location /SchedulerService-web>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>
</VirtualHost>
```

Example 3 `igdinternal_vh.conf` file

Contents of this file:

```
<VirtualHost WEBHOST1.example.com:7777>
    ServerName igdinternal.example.com:7777
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit

# WSM-PM
    <Location /wsm-pm>
    WLSRequest ON
    WebLogicCluster OIMHOST1.example.com:7010,OIMHOST2.example.com:7010
    WLProxySSL OFF
    WLProxySSLPassThrough OFF
    </Location>

    <Location /sodcheck>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
    </Location>

# OIM, role-sod profile
    <Location /role-sod>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

# Callback webservice for SOA. SOA calls this when a request is approved/rejected
# Provide the SOA Managed Server Port
    <Location /workflowservice>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
    </Location>

# used for FA Callback service.
    <Location /callbackResponseService>
    WLSRequest ON
    WLCookieName    oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

# spml xsd profile
    <Location /spml-xsd>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

# OIM, spml dsml profile
    <Location /spmlws>
    WLSRequest ON
    PathTrim /weblogic
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

    <Location /reqsvc>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
    </Location>

# SOA Infra
    <Location /soa-infra>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/OHS/component/oim_component.log"
    </Location>

# UMS Email Support
    <Location /ucs>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/OHS/component/oim_component.log"
    </Location>

    <Location /provisioning-callback>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

   <Location /CertificationCallbackService>
   WLSRequest ON
   WLCookieName oimjsessionid
   WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
   WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
   </Location>

   <Location /IdentityAuditCallbackService>
   WLSRequest ON
   WLCookieName oimjsessionid
   WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
   WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
   </Location>

# SOA Callback webservice for SOD - Provide the SOA Managed Server Ports
    <Location /soa/composer>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
    </Location>

    <Location /integration>
    SetHandler weblogic-handler
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLCookieName oimjsessionid
    </Location>
    <Location /sdpmessaging/userprefs-ui>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:8001,OIMHOST2.example.com:8001
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/soa_component.log"
    </Location>

    <Location /iam>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>
</VirtualHost>
```

Example 4 `prov_vh.conf`

Contents of this file:

```
<VirtualHost WEBHOST1.example.com:7777>
    ServerName https://prov.example.com:443
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit

    <Location /identity>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.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>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
    </Location>

    <Location /HTTPClnt>
    WLSRequest ON
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
    </Location>

# Requests webservice URL
    <Location /reqsvc>
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLProxySSL ON
    WLProxySSLPassThrough ON
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    </Location>

    <Location /FacadeWebApp>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
    </Location>

    <Location /iam>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
    </Location>

    <Location /OIGUI>
    SetHandler weblogic-handler
    WLCookieName oimjsessionid
    WebLogicCluster OIMHOST1.example.com:14000,OIMHOST2.example.com:14000
    WLLogFile "${ORACLE_INSTANCE}/diagnostics/logs/mod_wl/oim_component.log"
    WLProxySSL ON
    WLProxySSLPassThrough ON
    </Location>
</VirtualHost>
```

Example 5 `login_vh.conf`

Contents of this file:

```
<VirtualHost WEBHOST1.example.com:7777>
    ServerName https://login.example.com:443
    ServerAdmin you@your.address
    RewriteEngine On
    RewriteOptions inherit

#OAM Entries
  <Location /oam>
  WLSRequest ON
  WLProxySSL ON
  WLProxySSLPassThrough ON
  WLCookieName OAMJSESSIONID
  WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
  </Location>

  <Location /oamfed>
  WLSRequest ON
  WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
  WLCookieName OAMJSESSIONID
  WLProxySSL ON
  WLProxySSLPassThrough ON
  </Location>

# OAM Forgotten Password Page
  <Location /otpfp/>
  WLSRequest ON
  WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
  WLCookieName OAMJSESSIONID
  WLProxySSL ON
  WLProxySSLPassThrough ON
  </Location>

  <Location /ms_oauth>
  WLSRequest ON
  WebLogicCluster OAMHOST1.example.com:14100,OAMHOST2.example.com:14100
  WLCookieName OAMJSESSIONID
  WLProxySSL ON
  WLProxySSLPassThrough ON
  </Location>
</VirtualHost>
```

16 Configuring Oracle Access Management

You need to perform certain tasks in order to extend the enterprise deployment domain with the Oracle Access Management. This includes installing the Oracle Identity and Access Management, extending the domain for Oracle Access Management and completing post-configuration and verification tasks.

This chapter provides information on installing the Oracle Identity and Access Management, extending the domain for Oracle Access Management and completing post-configuration and verification tasks.

Variables Used in This Chapter

This topic lists the variables used in this chapter.

Variables

- *PRIMARY_OAM_SERVERS*
- *WEBGATE_TYPE*
- *ACCESS_GATE_ID*
- *OAM11G_OIM_WEBGATE_PASSWD*
- *COOKIE_DOMAIN*
- *COOKIE_EXPIRY_INTERVAL*
- *OAM11G_WG_DENY_ON_NOT_PROTECTED*
- *OAM11G_IDM_DOMAIN_OHS_HOST*
- *OAM11G_IDM_DOMAIN_OHS_PORT*
- *OAM11G_IDM_DOMAIN_OHS_PROTOCOL*
- *OAM11G_SERVER_LBR_HOST*
- *OAM11G_SERVER_LBR_PORT*
- *OAM11G_SERVER_LBR_PROTOCOL*
- *OAM11G_OAM_SERVER_TRANSPORT_MODE*
- *OAM_TRANSFER_MODE*
- *OAM11G_SSO_ONLY_FLAG*
- *OAM11G_IMPERSONATION_FLAG*
- *OAM11G_IDM_DOMAIN_LOGOUT_URLS*
- *OAM11G_OIM_INTEGRATION_REQ*
- *OAM11G_OIM_OHS_URL*
- *IDSTORE_PWD_OAMSOFTWAREUSER*
- *IDSTORE_PWD_OAMADMINUSER*
- *OAM11G_WLS_ADMIN_PASSWD*
- *IAD_MSERVER_HOME*
- *IAD_ASERVER_HOME*
- *WLS_AMA*
- *WebGate_IDM*
- *COMMON_IDM_PASSWORD*
- *WLS_OAM1*
- *WLS_AMA1*
- *WLS_OAM2*
- *WLS_AMA2*
- *JAVA_HOME*

- *OAM_PROXY_PORT*
- *IAD_HTTP_PORT*
- *IAD_ORACLE_HOME*

Configuring and Integrating with LDAP

This section describes how to configure and integrate Oracle Access Manager with LDAP.

This section contains the following topics:

- Setting a Global Passphrase
- Obtaining the Default Global Passphrase
- Configuring Access Manager to Use the LDAP Directory
- Adding WebGate Load Balancer Details
- Adding LDAP Groups to WebLogic Administrators

Setting a Global Passphrase

By default, Oracle Access Manager is configured to use the open security model. If you plan to change this mode using `idmConfigTool`, you must know the global passphrase. By default, Oracle creates a global passphrase for you. You can override this value, if required.

> **Note:**
>
> If you are using the latest 12c WebGate functionality by using OAP over REST calls, it is not important to change the security mode because REST calls do not use the OAP transport mode.

To set a global passphrase:

1. Log in to the OAM console using the URL, as the WebLogic Administration user (for example, weblogic):

   ```
   http://iadadmin.example.com/oamconsole
   ```

2. Click the **Configuration** tab.
3. Select **View**, and then **Access Manager** from the **Settings** launch pad.
4. Update the **Global Passphrase** with a value of your choice and make a note of it.
5. Enter the value you set as the **Global Passphrase**.
6. Click **Apply**.

Obtaining the Default Global Passphrase

If you prefer to use the default global passphrase, you can obtain it by completing the following steps:

1. Start WLST by using the following command:

   ```
   IAD_ORACLE_HOME/oracle_common/common/bin/wlst.sh
   ```

For example:

```
/u01/oracle/oracle_common/common/bin/wlst.sh
```

2. Connect to the domain using the command:

```
connect(WeblogicAdminUser',<WeblogicAdminPassword>','t3://
ADMINSERVER1:T3PORT'
```

For example:

```
connect('weblogic','<password>','t3://IADADMINVHN.example.com:7001')
```

3. Issue the WLST command:

```
displaySimpleModeGlobalPassphrase()
```

The system generated passphrase is displayed.

Configuring Access Manager to Use the LDAP Directory

After completing the initial installation and setting the security model, you have to associate Oracle Access Manager with the LDAP directory. You can use Oracle Unified Directory (OUD) as the LDAP directory.

To associate Access Manager and the LDAP directory, perform the following tasks:

- Creating a Configuration File
- Integrating Access Manager and LDAP Using the idmConfigTool
- Validating the OAM LDAP Configuration

Creating a Configuration File

Configuring Oracle Access Management to use LDAP requires running the `idmConfigTool` utility. Therefore, you must create a configuration file called `oam.props` to use during the configuration. The contents of this file will be the same as the Configuration file created in Creating a Configuration File with the following additions:

```
#IDSTORE PROPERTIES
IDSTORE_HOST: idstore.example.com
IDSTORE_PORT: 1389
IDSTORE_BINDDN: cn=oudadmin
IDSTORE_SEARCHBASE: dc=example,dc=com
IDSTORE_GROUPSEARCHBASE: cn=Groups,dc=example,dc=com
IDSTORE_USERNAMEATTRIBUTE: cn
IDSTORE_LOGINATTRIBUTE: uid
IDSTORE_USERSEARCHBASE: cn=Users,dc=example,dc=com
IDSTORE_SYSTEMIDBASE: cn=systemids,dc=example,dc=com
IDSTORE_NEW_SETUP: true
IDSTORE_DIRECTORYTYPE: OUD
IDSTORE_WLSADMINUSER: weblogic_iam
IDSTORE_WLSADMINGROUP: WLSAdministrators
IDSTORE_OAMADMINUSER: oamadmin
IDSTORE_OAMSOFTWAREUSER: oamLDAP
# OAM Properties
OAM11G_IDSTORE_NAME: OAMIDSTORE
```

```
PRIMARY_OAM_SERVERS: OAMHOST1.example.com:5575,OAMHOST2.example.com:5575
WEBGATE_TYPE: ohsWebgate12c
ACCESS_GATE_ID: Webgate_IDM
OAM11G_OIM_WEBGATE_PASSWD: Password
COOKIE_DOMAIN: .example.com
COOKIE_EXPIRY_INTERVAL: 120
OAM11G_WG_DENY_ON_NOT_PROTECTED: true
OAM11G_IDM_DOMAIN_OHS_HOST: login.example.com
OAM11G_IDM_DOMAIN_OHS_PORT: 443
OAM11G_IDM_DOMAIN_OHS_PROTOCOL: https
OAM11G_SERVER_LBR_HOST: login.example.com
OAM11G_SERVER_LBR_PORT: 443
OAM11G_SERVER_LBR_PROTOCOL: https
OAM11G_OAM_SERVER_TRANSFER_MODE: open
OAM_TRANSFER_MODE: open
OAM11G_SSO_ONLY_FLAG: false
OAM11G_IMPERSONATION_FLAG: false
OAM11G_IDM_DOMAIN_LOGOUT_URLS: /console/jsp/common/logout.jsp,/em/targetauth/
emaslogout.jsp
OAM11G_OIM_INTEGRATION_REQ: false
OAM11G_OIM_OHS_URL: https://prov.example.com:443/
# WebLogic Properties
WLSHOST: IADADMINVHN.example.com
WLSPORT: 7001
WLSADMIN: weblogic
```

OAM Property Descriptions:

- **OAM11G_IDSTORE_NAME** is the name you wish to assign to the ID store in OAM. This is an optional parameter.
- **PRIMARY_OAM_SERVERS** a comma-separated list of all of the OAM managed servers that are in the deployment. The format of this is Server Running the OAM Managed Server: OAM Proxy port. Note the proxy port used is not the OAM managed server listen port. The OAM Proxy port can be found in the worksheet (OAM_PROXY_PORT)
- **WEBGATE_TYPE** The type of webgate profile to create. This should always be `ohsWebgate12c`
- **ACCESS_GATE_ID** is the name of the Webgate Agent to create.
- **OAM11G_OIM_WEBGATE_PASSWD** is the password you wish to assign to the webgate agent you will be creating.
- **COOKIE_DOMAIN** is the domain you wish to associate the OAM cookie with this is normally the same as the *IDSTORE_SEARCH_BASE* in domain format. The search base can be found in the worksheet (REALM_DN).
- **COOKIE_EXPIRY_INTERVAL** the amount of time before a cookie is expired.
- **OAM11G_WG_DENY_ON_NOT_PROTECTED** this should always be set to true. It ensures that any attempt to access a resource not explicitly stated in the OAM Resource list will be rejected.
- **OAM11G_IDM_DOMAIN_OHS_HOST** this is the name of the Oracle HTTP Server (OHS) server which fronts the IAMAccessDomain. In the case of an enterprise deployment this will be the load balancer name.
- **OAM11G_IDM_DOMAIN_OHS_PORT** this is the port on which the OHS server fronting the IAMAccessDomain listens. In the case of an Enterprise Deployment, this will be the load balancer port. This is the IAD_HTTPS_PORT in the worksheet.
- **OAM11G_IDM_DOMAIN_OHS_PROTOCOL** this determines which process is being used when accessing the OHS server fronting the IAMAccessDomain.In the case of an

Enterprise Deployment this will be the load balancer protocol. In the Enterprise Deployment Blueprint SSL is terminated at the load balancer. But the URL will always have the HTTPS prefix, so this value should be set to `https`.

- **OAM11G_SERVER_LBR_HOST** this is the name of the virtual host configured on the load balancer for logging in. This is usually the same as **OAM11G_IDM_DOMAIN_OHS_HOST**.
- **OAM11G_SERVER_LBR_PORT** this is the port of the virtual host configured on the load balancer for logging in. This is usually the same as **OAM11G_IDM_DOMAIN_OHS_PORT**.
- **OAM11G_SERVER_LBR_PROTOCOL** this is the protocol of the virtual host configured on the load balancer for logging in. This is usually the same as **OAM11G_IDM_DOMAIN_OHS_PROTOCOL**.
- **OAM11G_OAM_SERVER_TRANSPORT_MODE** this is the type of OAM security transport to be used. This should be `Simple` for all platforms, except for AIX where it should be `Open`. You can specify `cert` if extra security is required. If you wish to use `cert`, refer to the Oracle Access Manager documentation for how to configure this.
- **OAM_TRANSFER_MODE** this is the type of OAM security transport to be used. This should be the same as **OAM11G_OAM_SERVER_TRANSPORT_MODE**
- **OAM11G_SSO_ONLY_FLAG** this is used to determine whether authentication mode is going to be used. For Enterprise Deployments this should be set to `false`.
- **OAM11G_IMPERSONATION_FLAG** determines whether OAM be configured for impersonation. Impersonation is typically used in help desk type applications where a support user "impersonates" and actual user for the purposes of providing support.
- **OAM11G_IDM_DOMAIN_LOGOUT_URLS** is a list of URLs that various products can invoke for the purposes of logging out.
- **OAM11G_OIM_INTEGRATION_REQ** If you are intending Oracle Identity Governance to handle forgotten password functionality then this parameter should be set to `true`. If you are using the new OAM forgotten password functionality then this value should be set to `false`.
- **OAM11G_OIM_OHS_URL** If you are planning on using OIM for Forgotten Password functionality then you need to specify the external entry point for OIG. This is the OIG URL to which OAM directs the requests. This url is made up of the following values from the worksheet:

  `https://prov.example.com:`*`IAG_HTTPS_PORT/`*

- **WLSHOST**: is the Admin Server listen address. For OAM configuration, this will be `IADADMINVHN.example.com`
- **WLSPORT**: is the Admin Server listen port. This is the IAD_WLS_PORT in the worksheet.
- **WLSADMIN** the user used to connect to the Admin Server

Integrating Access Manager and LDAP Using the idmConfigTool

This section describes how to integrate Oracle Access Manager and LDAP using the `idmConfigTool`.

> **Note:**
>
> Before running the `idmconfigTool`, ensure that the WLS_OAM1 and WLS_OAM2 Managed Servers are shut down.

Perform the following tasks on OAMHOST1:

1. Set the environment variables MW_HOME, JAVA_HOME and ORACLE_HOME.

   ```
   Set ORACLE_HOME to IAD_ORACLE_HOME/idm.
   MW_HOME to IAD_ORACLE_HOME
   ```

2. Run the idmConfigTool utility to perform the integration.

   The syntax of the command on Linux is:

   ```
   cd IAD_ORACLE_HOME/idm/idmtools/bin
   idmConfigTool.sh -configOAM input_file=configfile
   ```

   For example:

   ```
   idmConfigTool.sh -configOAM input_file=oam.props
   ```

   When the command runs you are prompted to enter the password of the account you are connecting to the Identity Store with. You are also asked to specify the passwords you want to assign to these accounts:

   - IDSTORE_PWD_OAMSOFTWAREUSER
   - IDSTORE_PWD_OAMADMINUSER
   - OAM11G_WLS_ADMIN_PASSWD

3. Check the log file for any errors or warnings and correct them. A file named automation.log is created in the directory where you run the tool.

4. Restart the Administration console.

> **Note:**
>
> After you run `idmConfigTool`, several files are created that you need for subsequent tasks. Keep these in a safe location.
>
> The following files exist in the following directory:
>
> ```
> IAD_ASERVER_HOME/output/Webgate_IDM
> ```
>
> You need these when you install the WebGate software.
>
> - cwallet.sso
> - ObAccessClient.xml
> - password.xml
> - aaa_cert.pem
> - aaa_key.pem

> **Note:**
>
> If the `WLS_AMA` servers were running when `configOAM` was run, then the `WebGate_IDM` artifacts may have been created in *`IAD_MSERVER_HOME`*`/output`. If this is the case, move them back to *`IAD_ASERVER_HOME`*`/output`.

Validating the OAM LDAP Configuration

To validate that this has completed correctly:

1. Access the OAM console using the following URL:

   ```
   http://iadadmin.example.com/oamconsole
   ```

2. Log in as the Access Manager administration user you created when you prepared the ID Store. For example `oamadmin`.
3. Click **Agents** from the Application Security screen.
4. When the Search SSO Agents screen appears, click **Search**.
5. You should see the Web Gate agent Webgate_IDM.
6. Log in to the WebLogic Administration Server Console as the default administrative user. For example, `weblogic`.
7. Click **Security Realms** on the left navigation pane.
8. On the Summary of Security Realms page, click **myrealm** under the **Realms** table.
9. On the Settings page for myrealm, go to the **Users and Groups** tab.

   > **Note:**
   >
   > The list of users and groups will be visible only after you restart the domain.

10. On to the users tab and check to see that LDAP users are displayed from your directory connector. For example: `OUDAuthenticator`.
11. On to the Groups tab and check to see that LDAP groups are displayed from your directory connector. For example: `OUDAuthenticator`.

Adding WebGate Load Balancer Details

In Oracle 12c, Oracle Webgate communicates with Oracle Access Manager 12c using the REST API calls rather than the traditional OAP calls. After running the `idmConfigTool`, you must manually update the WebGate Traffic Load Balancer details:

To update the details:

1. Access the OAM console using the following URL:

   ```
   http://iadadmin.example.com/oamconsole
   ```

2. Click **Configuration**.
3. On the launch pad, select **Access Manager** from the **Settings** box.
4. In the WebGate Traffic Load Balancer section, update the **OAM Server Host** and **OAM Server Port** to have the same values as those in the Load Balancing section on the page.
5. Click **Apply**.

Adding LDAP Groups to WebLogic Administrators

Oracle Access Manager requires access to the MBeans stored within the Administration Server. To enable the LDAP users to log in to the WebLogic Console and Fusion Middleware

Control, you must assign them the WebLogic administration rights. For Oracle Access Manager to invoke these Mbeans, users in the OAMAdministrators group must have the WebLogic administration rights.

When you implement single sign-on, you have to provide the LDAP group IDM administrators with the WebLogic administration rights to help them log in and perform the WebLogic administrative actions.

- Using the WebLogic Console

Using the WebLogic Console

To add the LDAP Groups `OAMAdministrators` and `WLSAdministrators` to the WebLogic Administrators:

1. Log in to the WebLogic Administration Server Console as the default administrative user. For example, `weblogic`.
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 **OAMAdministrators** in the **Group Argument** field and click **Add**.
11. Repeat for the Group **WLSAdministrators**.
12. Click **Finish** to return to the Edit Global Roles page.
13. The **Role Conditions** table now shows the groups **OAMAdministrators** or **WLSAdministrators** as role conditions.
14. Click **Save** to finish adding the Admin role to the OAMAdministrators and IDM Administrators Groups.

Updating WebGate Agents

When the `idmConfigTool` is run, it changes the default OAM security model and creates a new WebGate SSO Agent. However, it does not change the existing WebGate SSO Agents to the new security model. After running the `idmConfigTool`, you must update any WebGate agents that previously existed. This involves the following steps:

- Change the security mode to match that of the OAM servers. Failure to do so results in a security mismatch error.
- When WebGates are created at first install, they are unaware that a highly available (HA) installation is performed. After enabling HA, you must ensure that all of the OAM servers are included in the agent configuration, to ensure system continuity.

- You must check that any logout URLs are redirected to the hardware load balancer than one of the local OAM servers.
- Update the REST points for Oracle 12c WebGate HTTP OAM APIs.
- A WebGate agent called **IAMSuiteAgent** is created out of the box. This is created without any password protection and needs to have one added.

To perform these actions, complete the following steps:

1. Log in to the OAM Console at http://iadadmin.example.com/oamconsole using the OAM Administration user (oamadmin).
2. Click **Agents** pad on the Application Security screen.
3. Ensure that the **WebGates** tab is selected.
4. Click **Search**.
5. Click an Agent, for example: **IAMSuiteAgent**.
6. Set the Security value to the same value defined to **OAM Transfer Mode** on the Access Manager Configuration screen during response file creation.

   If you have changed the OAM security model using the `idmConfigTool`, change the security model used by any existing Webgates to reflect this change.

   Click **Apply**.
7. In the **Primary Server** list, click **+** and add any missing Access Manager Servers.
8. If a password has not already been assigned, enter a password into the **Access Client Password** field and click **Apply**.

   Assign an Access Client Password, such as the **Common IAM Password** (*`COMMON_IDM_PASSWORD`*) you used during the response file creation or an Access Manager-specific password, if you have set one.
9. Set **Maximum Connections** to 20. This is the total maximum number of connections for the primary servers, which is 10 x WLS_OAM1 connections plus 10 x WLS_OAM2 connections.
10. If you see the following in the **User Defined Parameters** or the **Logout redirect URL**:

    ```
    logoutRedirectUrl=http://OAMHOST1.example.com:14100/oam/server/logout
    ```

    Change it to:

    ```
    logoutRedirectUrl=https://login.example.com/oam/server/logout
    ```
11. If `OAMRestEndPointHostName` is present or missing for the WebGates `WebGate_11g` and `accessgate-oic`, ensure that it is set to: `login.example.com`.

    If `OAMRestEndPointPort` is present or missing for the WebGates `WebGate_11g` and `accessgate-oic`, ensure that it is set to: `443`.

    Without setting these two values, the 12c WebGate will not be able to use the new OAP REST APIs for authentication.
12. Click **Apply**.
13. Repeat Steps through for each WebGate.
14. Check that the security setting matches that of your Access Manager servers.

Updating Host Identifiers

When you access your domain you enter using different load balancer entry points. Each of these entry points (virtual hosts) need to be added to the Policy list. This ensures that if you request access to a resource using `login.example.com` OR `prov.example.com`, you have access to the same set of policy rules.

1. Access the OAM console at http://iadadmin.example.com/oamconsole.
2. Log in as the Access Manager administration user you created when you prepared the ID Store. For example `oamadmin`.
3. Select **Launch Pad** if not already displayed.
4. Click on **Host Identifiers** under **Access Manager**.
5. Click **Search**.
6. Click on **IAMSuiteAgent**.
7. Click **+** in the operations box.
8. Enter the following information.

Table 16-1 Host Name Port Values

Host Name	Port
`iadadmin.example.com`	80
`igdadmin.example.com`	80
`igdinternal.example.com`	7777
`prov.example.com`	443
`login.example.com`	443

9. Click **Apply**.

Adding Missing Policies to OAM

If any policies are missing, you have to add to ensure that Oracle Access Manager functions correctly.

You need to add the following additional policies:

Table 16-2 OAM Policy Information

Product	Resource Type	Host Identifier	Resource URL	Protection Level	Authentication Policy	Authorization Policy
ALL	HTTP	IAMSuiteAgent	`/consolehelp/**`	Excluded		
ALL	HTTP	IAMSuiteAgent	`/otpfp/**`	Excluded		
OIG	HTTP	IAMSuiteAgent	`/OIGUI/**`	Protected	Protected Higher Level Policy	Protected Resource Policy

Table 16-2 (Cont.) OAM Policy Information

Product	Resource Type	Host Identifier	Resource URL	Protection Level	Authentication Policy	Authorization Policy
OAM	HTTP	IAMSuiteAgent	`/iam/access/binding/api/v10/oap/**`	Excluded		
OAM	HTTP	IAMSuiteAgent	`/oam/services/rest/**`	Excluded		
OAM	HTTP	IAMSuiteAgent	`/iam/admin/config/api/v1/config/**`	Excluded		
OIG	HTTP	IAMSuiteAgent	`/iam/**`	Protected	Protected Higher Level Policy	Protected Resource Policy
OIG	HTTP	IAMSuiteAgent	`/iam/governance/**`	Excluded		
OIG	HTTP	IAMSuiteAgent	`/FacadeWebApp/**`	Protected	Protected Higher Level Policy	Protected Resource Policy
OIG	HTTP	IAMSuiteAgent	`/IdentityAuditCallbackService/**`	Excluded		
OIG	HTTP	IAMSuiteAgent	`/soa/composer`	Protected	Protected Higher Level Policy	Protected Resource Policy
OIG	HTTP	IAMSuiteAgent	`/soa-infra`	Protected	Protected Higher Level Policy	Protected Resource Policy
OIG	HTTP	IAMSuiteAgent	`/integration/**`	Protected	Protected Higher Level Policy	Protected Resource Policy
OUDSM	HTTP	IAMSuiteAgent	`/oudsm`	Excluded		

> **Note:**
>
> `/otpfp` is only required if you have implemented the OAM forgotten password functionality.

To add these policies:

1. Log in to the OAM Console at `http://iadadmin.example.com/oamconsole` using the `oamadmin` user.
2. From the Launch pad click **Application Domains** in the **Access Manager** section.
3. Click **Search** on the Search page.

A list of application domains appears.

4. Click the domain **IAM Suite**.
5. Click the **Resources** Tab.
6. Click **Create**.
7. Enter the information specified in the table above.
8. Click **Apply**.

Updating Federation Service Details

Now that Oracle Access Management (OAM) is configured, you must update the Federation services to access the Federation via the load balancer URL.

To do this:

1. Log in to the OAM Console at http://iadadmin.example.com/oamconsole.
2. Click **Configuration**.
3. In the settings pane, click **View**, and select **Federation** from the drop-down.
4. On the Federation Settings Page, update the **Provider ID** to `https://` *`login.example.com`*`/oam/fed`.
5. Click **Apply**.

Updating Idle Timeout Value

The default timeout value set in Access Manager is often too long and can cause issues such as, not logging a session out after that session has timed out. Therefore, it is recommended that this value is reduced to 15 minutes.

To update the idle timeout value:

1. Log in to the OAM Console at http://iadadmin.example.com/oamconsole.
2. Log in as the Access Manager administrator user you created during response file creation.
3. Click **Configuration**.
4. Select **Common Settings** under **Settings**.
5. Change **Idle Time out (minutes)** to `15`.
6. Click **Apply**.

Validating the Authentication Providers

Set the order of identity assertion and authentication providers in the WebLogic Server Administration console.

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. From the left navigation, select **Security Realms**.
4. Click the **myrealm** default realm entry.

5. Click the **Providers** tab.
6. From the table of providers, click the **DefaultAuthenticator**.
7. Set the Control Flag to `SUFFICIENT`.
8. Click **Save** to save the settings.
9. From the navigation breadcrumbs, click **Providers** to return to the list of providers.
10. Click **Reorder**.
11. Sort the providers to ensure that the OAM Identity Assertion provider is first and the DefaultAuthenticator provider is last.

Table 16-3 Sort order

Sort Order	Provider	Control Flag
1	OAMIDAsserter	`REQUIRED`
2	LDAP Authentication Provider	`SUFFICIENT`
3	DefaultIdentityAsserter	`N/A`
4	Trust Service Identity Asserter	`N/A`
5	DefaultAuthenticator	`SUFFICIENT`

12. Click **OK**.
13. Click **Activate Changes** to propagate the changes.
14. Shut down the Administration Server, Managed Servers, and any system components, as applicable.
15. Restart the Administration Server.
16. If you are going to configure ADF consoles with SSO, you can keep the managed servers down and restart them later. If not, you need to restart managed servers now.

Configuring Oracle ADF and OPSS Security with Oracle Access Manager

Some Oracle Fusion Middleware management consoles use Oracle Application Development Framework (Oracle ADF) security, which can integrate with Oracle Access Manager Single Sign-on (SSO). These applications can take advantage of Oracle Platform Security Services (OPSS) SSO for user authentication, but you must first configure the domain-level `jps-config.xml` file to enable these capabilities.

The domain-level `jps-config.xml` file is located in the following location after you create an Oracle Fusion Middleware domain:

```
ASERVER_HOME/config/fmwconfig/jps-config.xml
```

> **Note:**
>
> The domain-level `jps-config.xml` should not be confused with the `jps-config.xml` that is deployed with custom applications.

To update the OPSS configuration to delegate SSO actions in Oracle Access Manager, complete the following steps:

1. Change to the following directory:

   `ORACLE_COMMON_HOME/common/bin`

2. Start the WebLogic Server Scripting Tool (WLST):

   `./wlst.sh`

3. Connect to the Administration Server, by using the following WLST command:

   `connect(‘admin_user’,’admin_password’,’admin_url’)`

4. Run the `addOAMSSOProvider` command, as shown:

   ```
   addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
   logouturi="/oam/logout.html")
   ```

   The following table defines the expected value for each argument in the `addOAMProvider` command.

   > **Note:**
   >
   > Perform this action for each domain in your configuration.

Table 16-4 Expected Values for the Argument in the `addOAMProvider` command

Argument	Definition
loginuri	Specifies the URI of the login page **Note:** For ADF security enabled applications, "/*context-root*/adfAuthentication" should be provided for the 'loginuri' parameter. For example: `/${app.context}/adfAuthentication` **Note:** `${app.context}` must be entered as shown. At runtime, the application replaces the variable appropriately. Here is the flow: **a.** User accesses a resource that has been protected by authorization policies in OPSS, fox example. **b.** If the user is not yet authenticated, ADF redirects the user to the URI configured in *loginuri*. **c.** Access Manager, should have a policy to protect the value in *loginuri*: for example, "/*context-root*/ adfAuthentication". **d.** When ADF redirects to this URI, Access Manager displays a Login Page (depending on the authentication scheme configured in Access Manager for this URI).
logouturi	Specifies the URI of the logout page. The value of the *loginurl* is usually `/oam/logout.html`.
autologinuri	Specifies the URI of the autologin page. This is an optional parameter.

5. Disconnect from the Administration Server by entering the following command:

   ```
   disconnect()
   ```

6. Restart the Administration Server and the managed servers.

Starting the Managed Servers in the Domain

Start the Managed Servers in the following order:

- Starting the WLS_OAM1 Managed Server
- Starting the WLS_AMA1 Managed Server
- Starting the WLS_OAM2 Managed Server

- Starting the WLS_AMA2 Managed Server

Starting the WLS_OAM1 Managed Server

To start the WLS_OAM1 Managed Server:

1. Log in to the Oracle WebLogic Server Administration Console.

   `http://iadadmin.example.com/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**.
3. Verify that the server status is reported as `Running` in the Administration Console. If the server is shown as `Starting or Resuming`, wait for the server status to change to `Started`. If another status is reported (such as `Admin` or `Failed`), check the server output log files for errors.

Starting the WLS_AMA1 Managed Server

To start the WLS_AMA1 Managed Server:

1. Log in to the Oracle WebLogic Server Administration Console.

   `http://iadadmin.example.com/console`

2. Start the WLS_AMA1 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_AMA1**, and then click **Start**.
3. Verify that the server status is reported as `Running` in the Administration Console. If the server is shown as `Starting or Resuming`, wait for the server status to change to `Started`. If another status is reported (such as `Admin` or `Failed`), check the server output log files for errors.

Starting the WLS_OAM2 Managed Server

To start the WLS_OAM2 Managed Server:

1. Log in to the Oracle WebLogic Server Administration Console.

   `http://iadadmin.example.com/console`

2. Start the WLS_OAM2 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_OAM2**, and then click **Start**.
3. Verify that the server status is reported as `Running` in the Administration Console. If the server is shown as `Starting or Resuming`, wait for the server status to change to `Started`. If another status is reported (such as `Admin` or `Failed`), check the server output log files for errors.

Starting the WLS_AMA2 Managed Server

To start the WLS_AMA2 Managed Server:

1. Log in to the Oracle WebLogic Server Administration Console.

   ```
   http://iadadmin.example.com/console
   ```

2. Start the WLS_AMA2 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_AMA2**, and then click **Start**.
3. Verify that the server status is reported as `Running` in the Administration Console. If the server is shown as `Starting or Resuming`, wait for the server status to change to `Started`. If another status is reported (such as `Admin` or `Failed`), check the server output log files for errors.

Validating Access Manager

You can validate Access Manager by using the `oamtest` tool. To do this, perform the following steps:

1. Ensure that wls_oam managed server is up and running.
2. Ensure that JAVA_HOME is set in your environment by adding *JAVA_HOME*`/bin` to your path. For example:

   ```
   export PATH=$JAVA_HOME/bin:$PATH
   ```

3. Change the directory to the following:

   ```
   IAD_ORACLE_HOME/idm/oam/server/tester
   ```

4. Start the test tool in a terminal window using the command:

   ```
   java -jar oamtest.jar
   ```

5. When the OAM test tool starts, enter the following information in the Server Connection section of the page:
   - **Primary IP Address**: `OAMHOST1.example.com`
   - **Port**: `5575` (*`OAM_PROXY_PORT`*)
   - **Agent ID**: `Webgate_IDM`
   - **Agent Password**: `webgate password`
   - **Mode**: `Simple`

- **Global Passphrase**: Enter the value you set as the global password in Setting a Global Passphrase.

6. Click **Connect**.

   In the status window you'll see: `response] Connected to primary access server`.

7. In the Protected Resource URI section, enter the following information:

   - **Scheme**: http
   - **Host**: `iadadmin.example.com`
   - **Port**: `80` (*IAD_HTTP_PORT*)
   - **Resource**: `/oamconsole`

     Click **Validate**.

     In the status window you see: `[request] [validate] yes`.

8. In the User Identity window, enter:

   - **Username**: oamadmin
   - **Password**: oamadmin password
   - Click **Authenticate**.
   - In the status window, you see: `[request] [authenticate] yes`
   - Click **Authorize**.
   - In the status window you see. `[request] [authorize] yes`

Enabling Forgotten Password

This section describes how to set up the One Time Pin forgotten password functionality which is provided with Oracle Access Manager. If you want to configure the Challenge Question forgotten password functionality, as provided by Oracle Identity Governance, see Configuring and Integrating with LDAP and Integrating Oracle Identity Governance and Oracle Access Manager.

This section contains the following topics:

- Prerequisites for Enabling Forgotten Password
- Add Permissions to oamLDAP user
- Create an OTP Administrative Group in LDAP
- Enabling Adaptive Authentication Service
- Configuring Adaptive Authentication Plug-in
- Enabling Password Management in the Directory
- Storing User Messaging Credentials in CSF
- Setup for Forgot Password Link on Login Page
- Restarting the domain
- Validating the Forgotten Password Functionality
  If you have set up the OAM Forgotten Password functionality, rather than off-loading to OIM, you can validate the forgotten password using the `curl` command, which shows you the password policies in force.

Prerequisites for Enabling Forgotten Password

Forgotten Password Management in Oracle Access Manager takes the form of sending an Email or SMS message with a link to reset the password.

Email or SMS is sent using the Oracle User Messaging Service. Before enabling the Oracle Forgotten Password functionality, you first need to have an Oracle User Messaging deployment. This is often located inside the Oracle Governance Domain but can be located inside the Access Domain if that is all you are installing. Alternatively, it could be a completely independent domain.

Forgotten Password functionality works only if you have successfully configured Single Sign-On as described in Configuring Single Sign-On for an Enterprise Deployment.

Adding the User Messaging Service to the Access domain or creating a User Messaging Service domain is outside of the scope of the this EDG. For more information about installing and configuring the Oracle User Messaging Service, see Installing User Messaging Service and Configuring Oracle User Messaging Service in *Administering Oracle User Messaging Service*.

Add Permissions to oamLDAP user

When created out of the box the oamLDAP user (the user used to link OAM to LDAP) is granted privileges to read the LDAP directory. It is not however granted permission to update those users. You need to add these privileges for the OAM forgotten password functionality to work.

To do this you need to create an ldif file using your preferred text editor. This file will have the following content:

add_aci.ldif

```
dn: cn=oamLDAP,cn=systemids,dc=example,dc=com  changetype: modify add: ds-
privilege-name ds-privilege-name: password-reset


dn: cn=Users,dc=example,dc=com changetype: modify add: aci aci: (targetattr =
"*")(targetfilter= "(objectclass=inetorgperson)")(targetscope = "subtree")
(version 3.0; acl "iam admin changepwd"; allow
(compare,search,read,selfwrite,add,write,delete) userdn = "ldap:///
cn=oamLDAP,cn=systemids,dc=example,dc=com";)
```

Save the file.

On LDAPHOST1 action the file using the command:

```
OUD_ORACLE_INSTANCE/OUD/bin/ldapmodify -D cn=oudadmin -h LDAPHOST1 -p 1389 -f ./
add_aci.ldif
```

Create an OTP Administrative Group in LDAP

In order for the oamadmin group to be able to invoke forgotten password system calls it needs to be a member of the group **OTPRestUserGroup**. This group is not created by **idmConfigTool** and must therefore be created manually.

To do this you perform the following steps:

1. Create a file called **create_otp_group.ldif** with the following contents:

   ```
   dn: cn=OTPRestUserGroup,cn=Groups,dc=example,dc=com
   changetype: add
   objectClass: top
   objectClass: orclgroup
   objectClass: groupofuniquenames
   cn: OTPRestUserGroup
   description: Forgotten Password Admin group
   displayName: OTPRestUserGroup
   uniquemember: cn=oamadmin,cn=Users,dc=example,dc=com
   ```

2. Use the **ldapmodify** command to add the group to LDAP. For example:

   ```
   OUD_ORACLE_INSTANCE/OUD/bin/ldapmodify -D cn=oudadmin -h LDAPHOST1 -p 1389
   -f create_otp_group.ldif
   ```

Enabling Adaptive Authentication Service

Forgotten password requires the following service to be enabled.

To enable Adaptive Authentication Service, perform the following steps:

1. Log in to the Oracle Access Management Administration console as the `oamadmin` user, using the following URL:

   `http://iadadmin.example.com/oamconsole`

2. Click **Configuration**.
3. Click **Available Services**.
4. Click **Enable Service** next to Adaptive Authentication Service.
5. When prompted, confirm that you wish to enable the service.

Configuring Adaptive Authentication Plug-in

Now that the Authentication service is enabled, it needs to be informed about your User Messaging service.

To configure Adaptive Authentication Plug-In, perform the following steps:

1. Log in to the Oracle Access Management Administration console as the `oamadmin` user, using the following URL:

   `http://iadadmin.example.com/oamconsole`

2. From the Application Security Launch Pad, click **Authentication Plug-ins** in the Plug-ins panel. From the Authentication Plug-in tab, type **Adaptive** in the quick search box above the Plug-in Name column and hit **Enter**.

   The **AdaptiveAuthenticationPlugin** is displayed.

3. Enter the following plug in properties:

Table 16-5 AdaptiveAuthentication Plug-In Properties

Attribute	Value
UmsAvailable	True
UmsClientURL	Specify the entry point of your User Messaging service. If you have configured Oracle Identity Manager, then this will be:`http://igdinternal.example.com:7777/ucs/messaging/webservice`

4. Click **Save**.

Enabling Password Management in the Directory

By default OAM is not set to allow password management. This must be enabled through the OAM Console.

To enable Password Management in the Directory, perform the following steps:

1. Log in to the Oracle Access Management Administration console as the `oamadmin` user, using the following URL:

   http://*iadadmin.example.com*/oamconsole

2. Click **Configuration**.
3. Click **User Identity Stores**.
4. Click on your LDAP identity store in the OAM Identity Store section. For example, OAMIDSTORE
5. Click **Edit**
6. Select **Enable Password Management**.
7. Enter the details in the user information field.

Table 16-6 User Information Details

Attribute	Description
Global Common ID	Unique identifier in LDAP for the user for example: **uid**.
First Name	LDAP attribute which holds the users name, For example: **cn**.
Last Name	LDAP attribute which holds the users last name, For example: **sn**.
Email Address	This is the email address that will appear in the **From** section of sent emails.

8. Click **Apply**.

Storing User Messaging Credentials in CSF

Before you can access the User Messaging Service, you need to store the credentials in the WebLogic credential store.

To do this, execute the following set of WLST commands:

```
IAD_ORACLE_HOME/oracle_common/common/bin/wlst.sh
connect()
Please Enter your username: weblogic
Please Enter your password: COMMON_IDM_PASSWORD
Please enter your server URL [t3://localhost:7001] :t3://
IADADMINVHN.example.com:7001
You will now be connected to the domain. Execute the following commands:
createCred(map="OAM_CONFIG", key="umsKey", user="weblogic",
password="password")
createCred(map="OAM_CONFIG", key="oam_rest_cred", user="oamadmin",
password="password")
exit ()
```

The **umsKey** is used to provide the credentials to the unified messaging server which will send out your email or sms notifications.

The **oam_rest_cred** is the user allowed to invoke the Rest services in the OAM server.

In the above commands, `weblogic` is the domain administrative user, and `password` is its associated password.

Setup for Forgot Password Link on Login Page

The following REST API command enables the OTP forgot password link on the default login page in OAM.

```
 curl -X PUT \
  https://login.example.com/oam/services/rest/access/api/v1/config/
otpforgotpassword/ \
  -u oamadmin:Password \
  -H 'content-type: application/json' \
  -d
'{"displayOTPForgotPassworLink":"true","defaultOTPForgotPasswordLink":"false",
"localToOAMServer":"true","forgotPasswordURL":"https://login.example.com/
otpfp/pages/fp.jsp", "mode":"userselectchallenge"}'
```

Enter the required attributes and values:

Table 16-7 Forgot Password Link on Login Page

Attributes	Value
base_url	Main entry point of OAM. For example,https://login.example.com

Table 16-7 (Cont.) Forgot Password Link on Login Page

Attributes	Value
mode	distribution_mode The distribution mode determines how the password reset url is sent to the end user. Valid values are: **email, sms, userchoose, userselectchallenge**. The last entry allows the user to choose from masked values. • **Email** -- OTP will be sent to the email configured in the mail field. • **SMS** -- OTP will be sent to the mobile number configured in the mobile field. • **Userchoose** -- OTP will be sent by letting the user choose either the email or the mobile option, without the exact values. • **Userselectchallenge** -- User can see the masked values either as email or the mobile and select one of the options.

Note:

If you are using self signed certificates in the load balancer the curl command may object with a message similar to:
curl performs SSL certificate verification by default, using a **bundle** of Certificate Authority (CA) public keys (CA certs). If the default bundle file isn't adequate, you can specify an alternate file using the --cacert option. If this HTTPS server uses a certificate signed by a CA represented in the bundle, the certificate verification probably failed due to a problem with the certificate (it might be expired, or the name might not match the domain name in the URL). If you like to turn off curl's verification of the certificate, use the **-k (or --insecure)** option.

If you see this message and are sure, add **-k after -u oamadmin:Password**.

Verify that this has succeeded by accessing the followig URL in a browser:

```
https://login.example.com/oam/services/rest/access/api/v1/config/
otpforgotpassword
```

When prompted, enter your `oamadmin` account and password.

Note:

One of the OAM managed servers must be running for this command to succeed.

Restarting the domain

Shutdown and restart the **Administration Server** and all of the managed servers (WLS_AMA1, WLS_AMA2, WLS_OAM1, WLS_OAM2).

Validating the Forgotten Password Functionality

If you have set up the OAM Forgotten Password functionality, rather than off-loading to OIM, you can validate the forgotten password using the `curl` command, which shows you the password policies in force.

To validate the Forgotten Password functionality, run the following `curl` command:

```
curl -X GET https://login.example.com/oam/services/rest/access/api/v1/pswdmanagement/
UserPasswordPolicyRetriever/oamadmin?description=true  -u oamadmin:<password> -k
```

This command displays the password policies.

If this command works, access the protected URL listed below. After you enable single sign-on, you see a link for the forgotten password on the login page. Click this link and enter the user name for which you want to reset the password. Click **Generate Pin** to receive an email, which enables you to change the password.

```
http://iadadmin.example.com/console
```

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see Performing Backups and Recoveries for an Enterprise Deployment.

17
Configuring Oracle Identity Governance

Configuration of Oracle Identity Governance (OIG) comprises a series of steps, including integrating OIG with Oracle SOA suite, configuring the web tier, integrating OAM and OIG, configuring OIG workflow notifications, and so on. In the end, you back up the configuration.

This chapter includes the following topics:

- Update the SOA Integration URL
- Configuring OIM Workflow Notifications to be Sent by Email
  OIM uses the human workflow, which is integrated with the SOA workflow. The SOA server configures email to receive the notifications that are delivered to the user mailbox. The user can accept or reject the notifications.
- Adding the wsm-pm Role to the Administrators Group
  After you configure a new LDAP-based Authorization Provider and restart the Administration Server, add the enterprise deployment administration LDAP group (OIMAdministrators) as a member to the `policy.Updater` role in the `wsm-pm` application stripe.
- Adding the Oracle Access Manager Load Balancer Certificate to the Oracle Keystore Service
  The Oracle Identity Governance to Business Intelligence Reports link inside of the Self Service application requires that the SSL certificate used by the load balancer be added to the Oracle Keystore Service Trusted Certificates.
- Restarting the IAMGovernanceDomain
- Setting Challenge Questions
  If you have integrated OAM and OIM, then after the environment is ready, you need to set up the challenge questions for your system users.
- Integrating Oracle Identity Manager with Oracle Business Intelligence Publisher
  Oracle Identity Manager comes with a number of prebuilt reports that can be used to provide information about Oracle Identity and Access Management.

Variables Used When Configuring Oracle Identity Governance

While configuring Oracle Identity Governance, you will reference the directory variables listed in this section.

The values for several directory variables are defined in File System and Directory Variables Used in This Guide.

- *IGD_ORACLE_HOME*
- *IGD_ASERVER_HOME*
- *IGD_MSERVER_HOME*
- *APPLICATION_HOME*
- *DEPLOY_PLAN_HOME*
- *JAVA_HOME*
- *DOMAIN_HOME*
- *IDSTORE_DIRECTORYTYPE*
- *IDSTORE_SEARCHBASE*
- *IDSTORE_USERSEARCHBASE*
- *IDSTORE_GROUPSEARCHBASE*
- *IDSTORE_OIMADMINUSERDN*
- *IDSTORE_OIMADMINUSER_PWD*
- *IDSTORE_EMAIL_DOMAIN*
- *OIM_HOST*

- *OIM_PORT*
- *WLS_OIM_SYSADMIN_USER*
- *WLS_OIM_SYSADMIN_USER_PWD*
- *OIM_WLS_HOST*
- *OIM_WLS_PORT*
- *OIM_WLS_ADMIN*
- *OIM_SERVER_NAME*
- *WL_HOME*
- *OAM_HOST*
- *OAM_PORT*
- *ACCESS_SERVER_HOST*
- *ACCESS_SERVER_PORT*
- *ACCESS_GATE_ID*
- *SSO_ACCESS_GATE_PASSWORD*
- *COOKIE_DOMAIN*
- *OAM_TRANSFER_MODE*
- *OIM_LOGINATTRIBUTE*
- *OAM11G_WLS_ADMIN_HOST*
- *OAM11G_WLS_ADMIN_PORT*
- *OIM_WLSHOST*
- *OIM_WLSPORT*
- *OIM_WLSADMIN*
- *OIM_WLSADMIN_PWD*
- *OIM_SERVER_NAME*
- *IDSTORE_OAMADMINUSER*
- *IDSTORE_OAMADMINUSER_PWD*
- *OAM11G_WLS_ADMIN_USER*
- *OAM11G_WLS_ADMIN_PASSWD*
- *IDSTORE_HOST*
- *IDSTORE_PORT*
- *IDSTORE_BINDDN*
- *IDSTORE_BINDPWD*

In addition, you'll be referencing the following virtual IP (VIP) address defined in Reserving the Required IP Addresses for an Enterprise Deployment:

- ADMINVHN

Actions in this chapter will be performed on the following host computers:

- OIMHOST1
- OIMHOST2

- WEBHOST1
- WEBHOST2

Starting and Validating the Oracle Identity Governance Managed Servers

Now that you have extended the domain, started the Administration Server, and propagated the domain to the other hosts, you can start the newly configured Oracle Identity Governance Managed Servers.

This process involves three tasks as described in the following sections.

- Starting the Oracle Identity Governance Managed Servers and Bootstrapping the Domain
  Unlike previous releases you no longer need to run the Oracle Identity Governance configuration wizard to deploy the OIM artifacts into the domain. However, you are required to boot strap the domain. This automatically performs many of the actions that used to be performed by the OIM configuration wizard in previous releases.
- Starting the WLS_SOA1 and WLS_OIM1 Managed Servers
- Validating the Managed Server by Logging in to the Identity Console
- Starting and Validating WLS_SOA2, WLS_OIM2, and WLS_WSM2 Managed Servers
  After validating the successful configuration and startup of the WLS_SOA1 and WLS_OIM1 Managed Servers, you can start and validate the WLS_SOA2, WLS_OIM2, and WLS_WSM2 Managed Servers.

Starting the Oracle Identity Governance Managed Servers and Bootstrapping the Domain

Unlike previous releases you no longer need to run the Oracle Identity Governance configuration wizard to deploy the OIM artifacts into the domain. However, you are required to boot strap the domain. This automatically performs many of the actions that used to be performed by the OIM configuration wizard in previous releases.

Bootstrapping the domain is largely automatic and is performed by starting and stopping the managed servers in the domain in the following order:

1. Start the Oracle SOA Suite Managed Server WLS_SOA1.
2. Start the Oracle Identity Governance Managed Server WLS_OIM1.

   The bootstrap process starts the Managed Server, and then stops it again automatically. You may see a Failed status in the WebLogic console, which can be ignored.
3. Stop the Oracle SOA Suite Managed Server WLS_SOA1.
4. Stop WLS_OIM1.
5. Stop the WebLogic Administration Server.
6. Start the WebLogic Administration Server.
7. Start the Oracle SOA Suite Managed Servers WLS_SOA1 and WLS_SOA2.
8. Start the Oracle Identity Governance Managed Servers WLS_OIM1 and WLS_OIM2.

In order for the bootstrapping process to successfully complete, it must occur when the OIM server is started from the *IGD_ASERVER_HOME* directory. However, the Node Manager that runs out of the *IGD_ASERVER_HOME* communicates using the `igdadmin` address. Rather than

temporarily reconfiguring the Managed Servers to use this address, the Managed Servers can be started outside of Node Manager for the bootstrap process. Once the process is complete, the Managed Servers will be moved to local storage and Node Manager configured will be able to start and stop them.
To start the Managed Servers without Node Manager, you must run the following command from the directory `IGD_ASERVER_HOME/bin`:

- Command for starting the Oracle SOA Suite Managed Server: `./startManagedWeblogic.sh WLS_SOA1`
- Command for starting the Oracle Identity Governance Managed Server: `./startManagedWeblogic.sh WLS_OIM1`

When you execute these commands, you will be prompted to enter the WebLogic username and password. These commands run interactively, that is, after starting a Managed Server, control will not be returned to the command line. This does not matter as it is a one time operation.

> **Note:**
>
> You cannot perform these actions using Node Manager at this time.

Starting the WLS_SOA1 and WLS_OIM1 Managed Servers

To start the WLS_SOA1 and WLS_OIM1 Managed Servers:

1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

   ```
   http://igdadmin.example.com/em
   ```

   > **Note:**
   >
   > If you have already configured Web tier, use `http://igdadmin.example.com/em`.

2. Log in to Fusion Middleware Control using the Administration Server credentials.
3. In the **Target Navigation** pane, expand the domain to view the Managed Servers in the domain.
4. Select only the **WLS_WSM1** Managed Server and click **Start Up** on the Oracle WebLogic Server toolbar.
5. When the startup operation is complete, navigate to the Domain home page and verify that the **WLS_WSM1** Managed Server is up and running.
6. Start the managed servers one after the other. Ensure one is started and then start the next one. Repeat for the servers WLS_SOA1 and WLS_OIM1.

Validating the Managed Server by Logging in to the Identity Console

Validate the Oracle Identity Manager Server instance by bringing up the Oracle Identity Manager Console in a Web browser at:

```
http://OIMHOST1.example.com:14000/identity/
```

```
http://OIMHOST11.example.com:14000/sysadmin/
```

Log in using the **xelsysadm** username and password.

Validate the SOA configuration.

```
http://OIMHOST1.example.com:8001/soa-infra
```

Starting and Validating WLS_SOA2, WLS_OIM2, and WLS_WSM2 Managed Servers

After validating the successful configuration and startup of the WLS_SOA1 and WLS_OIM1 Managed Servers, you can start and validate the WLS_SOA2, WLS_OIM2, and WLS_WSM2 Managed Servers.

To start and validate the WLS_SOA2 Managed Server, use the procedure in Starting and Validating the WLS_SOA1 Managed Serverfor WLS_SOA2 Managed Server. Use the procedure to start and validate the WLS_OIM2 and WLS_WSM2 Managed Servers too.

For the validation URL, enter the following URL in your web browser and log in using the enterprise deployment administrator user:

For Static cluster:

```
http://OIMHOST2:14000/identity
```

For Dynamic cluster:

```
http://OIMHOST2:14001/identity
```

Analyzing the Bootstrap Report

When you start the Oracle Identity Governance server, the bootstrap report is generated at `$IGD_ASERVER_HOME/servers/WLS_OIM1/logs/BootStrapReportPreStart.html`.

The bootstrap report `BootStrapReportPreStart.html` is an html file that contains information about the topology that you have deployed, the system level details, the connection details like the URLs to be used, the connectivity check, and the task execution details. You can use this report to check if the system is up, and also to troubleshoot the issues, post-configuration. Every time you start the Oracle Identity Governance server, the bootstrap report is updated.

Sections in the Bootstrap Report

- **Topology Details**

  This section contains information about your deployment. It shows whether you have configured a cluster setup, SSL enabled, or upgraded an Oracle Identity Manager environment from 11*g* to 12*c*.

- **System Level Details**

  This section contains information about the JDK version, Database version, JAVA_HOME, DOMAIN_HOME, OIM_HOME, and MIDDLEWARE_HOME.

- **Connection Details**

  This section contains information about the connect details like the Administration URL, OIM Front End URL, SOA URL, and RMI URL.

This also shows whether the Administration Server, Database, and SOA server is up or not.

- **Execution Details**

  This section lists the various tasks and their statuses.

Validating the Fusion Middleware Control Application

After the bootstrap process has been executed and validated, access to the Fusion Middleware Control application should be available.

To navigate to the Fusion Middleware Control application, enter the following URL, and log in with the Oracle WebLogic Server administrator credentials:

```
http://IGDADMINVHN.example.com:7101/em
```

Configuring the Web Tier for the Domain

Configure the web server instances on the web tier so that the instances route requests for both public and internal URLs to the proper clusters in the extended domain.

For additional steps in preparation for possible scale-out scenarios, see Updating Cross Component Wiring Information.

- Integrating Oracle Identity Governance with Oracle SOA Suite
  Use the Enterprise Manager console to integrate Oracle Identity Governance with Oracle SOA Suite.
- Validating the Oracle SOA Suite URLs Through the Load Balancer

Integrating Oracle Identity Governance with Oracle SOA Suite

Use the Enterprise Manager console to integrate Oracle Identity Governance with Oracle SOA Suite.

To integrate Oracle Identity Governance with Oracle SOA suite:

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

   ```
   http://igdadmin.example.com/em
   ```

   or

   ```
   http://IGDADMINVHN.example.com:7101/em
   ```

   The Administration Server host and port number were in the URL on the End of Configuration screen (Writing Down Your Domain Home and Administration Server URL). The default Administration Server port number is `7101`.

   The login credentials were provided on the Administrator Account screen (Configuring the Administrator Account).

2. Click **weblogic_domain**, and then click **System Mbean Browser**.
3. In the search box, enter `OIMSOAIntegrationMBean`, and click **Search**. The mbean is displayed.

> **Note:**
>
> If Oracle Identity Governance still starting (coming up) or is just started (RUNNING MODE), the Enterprise Manager does not show any Mbeans defined by OIM. 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. Go to the **Operations** tab of mbean, and select **integrateWithSOAServer**.
5. Enter the following information:
   - **Weblogic Administrator User Name**: Enter the name of the WebLogic domain administrator account. For example, `weblogic`.
   - **Weblogic Administrator Password**: Enter the password for the above account.
   - **OIM Front end URL**: Set this to the load balancer virtual host used for internal call backs. For example:
     `http://igdinternal.example.com:7777/`
   - **OIM External Front End URL**: Set this URL to the main load balancer virtual host used for Oracle Identity Governance. For example:
     `https://prov.example.com:443/`
   - **SOA SOAP URL**: Set this URL to the load balancer virtual host used for internal call backs. For example:
     `http://igdinternal.example.com:7777/`
   - **SOA RMI URL**: Set this URL to the load balancer virtual host used for internal call backs. For example:
     `http://igdinternal.example.com:7777/`
   - **UMS Webservice URL**: Set this URL to the load balancer virtual host used for internal call backs. For example:
     `http://igdinternal.example.com:7777/ucs/messaging/webservice`
6. Click **Invoke**.

Validating the Oracle SOA Suite URLs Through the Load Balancer

To validate the configuration of the Oracle HTTP Server virtual hosts and to verify that the hardware load balancer can route requests through the Oracle HTTP Server instances to the application tier:

1. Verify that the server status is reported as **Running** in the Administration Console.

   If the server is shown as **Starting** or **Resuming**, wait for the server status to change to **Started**. If another status is reported (such as **Admin** or **Failed**), check the server output log files for errors.
2. Verify that you can access these URLs:

> **Note:**
>
> It is not necessary at this stage to attempt to login to the individual pages. All you are checking is that the pages can be accessed through the load balancer and the web server.

- `http://igdinternal.example.com:7777/soa-infra`
- `http://igdinternal.example.com:7777/integration/worklistapp`
- `http://igdinternal.example.com:7777/soa/composer`

Managing the Notification Service

An event is an operation that occurs in Oracle Identity Manager, such as user creation, request initiation, or any custom event created by the user. These events are generated as part of the business operations or through the generation of errors. Event definition is the metadata that describes the event.

To define the metadata for events, you must identify all event types supported by a functional component. For example, as a part of the scheduler component, metadata is defined for a scheduled job execution failure and shutting down of the scheduler. Every time a job fails or the scheduler shuts down, the associated events get triggered, and the notifications associated with the event get sent.

The data available in the event is used to create the content of the notification. The different parameters defined for an event help the system to select the appropriate notification template. The various parameters defined for an event help the system decide which event variables should be made available at template design time.

A notification template is used to send notifications. These templates contain variables that refer to available data to provide more context to the notifications. The notification is sent through a notification provider. Examples of such channels are e-mail, Instant Messaging (IM), Short Message Service (SMS), and voice. To use these notification providers, Oracle Identity Manager uses Oracle User Messaging Service (UMS).

At the back end, the notification engine is responsible for generating the notification and utilizing the notification provider to send the notification.

- Using SMTP for Notification
- Adding a CSF Key

Using SMTP for Notification

Using SMTP for notification involves configuring the SMTP email notification provider properties and adding the CSF key.

Configuring the SMTP Email Notification Provider Properties

To configure SMTP Email Notification Provider properties by using the **EmailNotificationProviderMBean MBean** :

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

   ```
   http://igdadmin.example.com/em
   ```

   or

   ```
   http://igdadmin.example.com:7101/em
   ```

The Administration Server host and port number were in the URL on the End of Configuration screen (Writing Down Your Domain Home and Administration Server URL). The default Administration Server port number is `7001`.

The login credentials were provided on the Administrator Account screen (Configuring the Administrator Account).

2. Click **weblogic_domain**, and then click **System Mbean Browser**.
3. In the search box, enter `EmailNotificationProviderMBean`, and click **Search**. The mbean is displayed.

> **Note:**
>
> If Oracle Identity Governance still starting (coming up) or is just started (RUNNING MODE), the Enterprise Manager does not show any Mbeans defined by OIM. 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. Ensure that the correct information is entered for your email server in particular:

Table 17-1 SMTP Email Notification Provider Properties

Attribute	Value
CSFKey	Set this to a name of a CSF credential, this can be any name and will be used while adding a CSF key. For example; **mailUser**
Enabled	Set to true.
MailServerName	Set to the host name of your email server.
WSUrl	`http://igdinternal.example.com/ucs/messaging/webservice`

5. Click **Apply** to save the changes.

Adding a CSF Key

To add a CSF key:

1. Login to Oracle Enterprise Manager.
2. Click WebLogic Domain and select **Security>Credentials**.
3. Expand **oracle.wsm.security** and click **Create Key**.
4. Enter the following information.

Table 17-2 CSF Key Properties

Attribute	Value
Key name	Enter the value of the credential Key, this must be the same value as defined in Using SMTP for Notification for example; mailUser.
Username	Enter the name of the user you use to authenticate with your email server.

Table 17-2 (Cont.) CSF Key Properties

Attribute	Value
Password/Confirm Password	Enter the password of the user you use to authenticate with your email server.
Description	Provide a description of the key being created. For example, Mail Server Credentials

5. Click **OK**.

Configuring the Messaging Drivers

Each messaging driver needs to be configured. You have to configure this service if you want to enable OAM's forgotten password functionality.

- Configuring the Email Driver

Configuring the Email Driver

To configure the driver to send and emails then you need to perform the following steps:

1. Log in to the Oracle Fusion Middleware Control.
2. Click the **Target Navigation** icon next to the Domain name.
3. Click **usermessagingserver (WLS_SOA1)** under User Messaging Service. A list of all the drivers will be shown.
4. Click **Configure Driver** next to the User Messaging Email Driver.
5. If a configuration does not exist then click **Create**. If the configuration exists, click **Edit**.
6. Update the attributes with the required details.

Table 17-3 Configuring the Email Driver Attributes

Attributes	Values
Name	MyemailServer
Sender Address	Enter the From email address for the emails you wish to send in the format: EMAIL:myuser@example.com
Capability	Choose whether you are going to send or receive emails. Complete the following Email Properties using the values specific to your organisation. Contact your email administrator for details, the details below are for Sending only. Refer to the documentation for receiving email details. • Outdoing Mail server. • Outgoing Mail server port • Outgoing email Server Security • Outgoing User name and password, if your email server requires it.

7. Click **Test** to validate the information.
8. Click **OK** to save the information.

Increasing Database Connection Pool Size

The default database connection pool size needs to be increased when Oracle Identity Governance is used in conjunction with a connector that allows interactions with an LDAP directory.

To do this, complete the following steps:

1. Log in to the WebLogic Server Administration Console in `IAMGovernanceDomain`.
2. Click **Lock & Edit**.
3. Click **Services** and then click **Data Sources**.
4. Click the data source **mds-oim**.
5. Go to the **Connection Pool** tab.
6. Modify the following properties with the values specified:
   - Initial Capacity: 50
   - Maximum Capacity: 150
   - Minimum Capacity: 50
   - Inactive Connection Timeout value to 30 from any other value

   > **Note:**
   >
   > Inactive Connection Timeout is in the Advanced section.

7. Click **Save**.
8. Click **Activate Changes**.
9. You will receive a message **All changes have been activated. No restarts are necessary**.

Forcing Oracle Identity Governance to use Correct Multicast Address

Oracle Identity Governance uses multicast for certain functions. By default, the managed servers communicate using the multi cast address assigned to the primary host name. If you wish multicast to use a different network, for example, of the internal network, you must complete the following additional steps:

1. Log in to the WebLogic Administration console using the following URL:

   `http://IGDADMIN.example.com/console`

2. Under **Domain Structure**, click **Environment** and then expand **Servers**. The Summary of Servers page is displayed.
3. Click **Lock & Edit**.
4. Click the OIM Managed Server name, for example, `WLS_OIM1` on the list of servers. The Settings for WLS_OIM1 are displayed.
5. Go to the **Server Start** tab.

6. Add the following line to the arguments field:

   ```
   -Dmulticast.bind.address=OIMHOST1
   ```

7. Click **Save**.
8. Repeat for the Managed Server `WLS_OIM2`. When doing so, make sure you add the following line to the arguments field:

   ```
   -Dmulticast.bind.address=OIMHOST2
   ```

9. Click **Activate Changes** and restart the managed servers `WLS_OIM1` and `WLS_OIM2`.

Integrating Oracle Identity Governance with LDAP

Integrating Oracle Identity Governance includes the following topics:

- Installing the Connector Bundle
- Configuring the Oracle Connector for LDAP
- Add Missing Object Classes

Installing the Connector Bundle

1. Download the Connector bundle from the artifactory: Download Connector Bundle
   - For OID or OUD, download the Connector bundle corresponding to Oracle Internet Directory.

   > **Note:**
   >
   > For all directory types, the required Connector version for OIG-OAM integration is 12.2.1.3.0.

2. Unzip the Connector bundle to the desired connector path under `$ORACLE_HOME/idm/server/ConnectorDefaultDirectory`.

   For example:

   ```
   $IGD_ORACLE_HOME/idm/server/ConnectorDefaultDirectory
   ```

Configuring the Oracle Connector for LDAP

The Oracle Connector for LDAP allows you to store users and passwords in a certified LDAP directory. Configure the connector before using it. Perform the following steps to configure the connector:

1. Change directory to *`IGD_ORACLE_HOME`*`/idm/server/ssointg/config`.
2. Edit the file `configureLDAPConnector.config` shown below:

   ```
   ##-----------------------------------------------------------##
   ## [configureLDAPConnector]
   IDSTORE_DIRECTORYTYPE=OUD
   IDSTORE_HOST=idstore.example.com
   IDSTORE_PORT=1389
   IDSTORE_BINDDN=cn=oudadmin
   IDSTORE_OIMADMINUSERDN=cn=oimLDAP,cn=systemids,dc=example,dc=com
   ```

```
IDSTORE_SEARCHBASE=dc=example,dc=com
IDSTORE_USERSEARCHBASE=cn=Users,dc=example,dc=com
IDSTORE_GROUPSEARCHBASE=cn=Groups,dc=example,dc=com
IDSTORE_USERSEARCHBASE_DESCRIPTION=Default user container
IDSTORE_GROUPSEARCHBASE_DESCRIPTION=Default group container
IDSTORE_EMAIL_DOMAIN=example.com
OIM_HOST=OIMHOST1.example.com
OIM_PORT=14000
WLS_OIM_SYSADMIN_USER=xelsysadm
OIM_WLSHOST=IGDADMINVHN.example.com
OIM_WLSPORT=7101
OIM_WLSADMIN=weblogic
OIM_SERVER_NAME=oim_server1
CONNECTOR_MEDIA_PATH=IGD_ORACLE_HOME/idm/server/ConnectorDefaultDirectory/
OID-12.2.1.3.0
```

Note:

You can also specify the passwords directly in the file, if required. If you do not specify the passwords, you will be prompted for them at runtime.

Parameters are:

- `OIM_WLSADMIN_PWD`
- `IDSTORE_BINDDN_PWD`
- `WLS_OIM_SYSADMIN_USER_PWD`
- `ADMIN_USER_PWD`
- `IDSTORE_OIMADMINUSER_PWD`

Save the file when done.

This table lists the properties of configuring the LDAPConnector.

Table 17-4 Configure LDAPConnector Properties

Attribute	Description
IDSTORE_HOST	It is the Load Balancer name for the LDAP directory for example: idstore.example.com
IDSTORE_PORT	It is the LDAP port on the load balancer for example 1389 for OUD.
IDSTORE_DIRECTORYTYPE	It is the type of LDAP directory you are using OUD.
IDSTORE_BINDDN	It is the credential used to connect to the directory to perform administrative actions, for example, oudadmin for OUD.
IDSTORE_SEARCHBASE	It is the root directory tree in the directory.
IDSTORE_USERSEARCHBASE	It is the location in the directory where users are stored.
IDSTORE_GROUPSEARCHBASE	It is the location in the directory where groups are stored.

Table 17-4 (Cont.) Configure LDAPConnector Properties

Attribute	Description
IDSTORE_OIMADMINUSERDN	It is the name of the user that OIM will use to connect to LDAP.
IDSTORE_EMAIL_DOMAIN	It is the email domain.
OIM_HOST	This the the hostname that the OIM Managed server WLS_OIM1 is listening on, for example OIMHOST1.
OIM_PORT	It is the port number of the WLS_OIM1 managed server.
WLS_OIM_SYSADMIN_USER	It is the OIM administrator account for example xelsysadm.
OIM_WLSHOST	It is the listen address of the IAMGovernanceDomain administration server, for example IGDADMINVHN
OIM_WLSPORT	It is the administration servers port for example 7101.
OIM_WLSADMIN	It is the name of the IAMGovernance Domain administration user. For example weblogic.
CONNECTOR_MEDIA_PATH	It is the location where you have installed the connector.
OIM_SERVER_NAME	It is the name of the OIM Managed server that is running. For example; wls_oim1.

> **Note:**
>
> You should use the same values as you specified for these parameters in Creating a Configuration File.

3. Locate the properties file, `ssointg-config.properties`, available at *`IGD_ORACLE_HOME/idm/server/ssointg/config/`* and set the **configureLDAPConnector** value to true. All other values should be set to false.

```
##----------------------------------------------------------##

generateIndividualConfigFiles=false
prepareIDStore=false
configOAM=false
addMissingObjectClasses=false
populateOHSRules=false
configureWLSAuthnProviders=false
configureLDAPConnector=true
## configureLDAPConnector takes care of updating container rules
## Additional option is provided in case rules need to be updated again
updateContainerRules=false
configureSSOIntegration=false
enableOAMSessionDeletion=false
```

4. Execute the script OIGOAMIntegration for configuring the connector.

5. For example:

   ```
   cd IGD_ORACLE_HOME/idm/server/ssointg/bin
   export JAVA_HOME=JAVA_HOME
   export ORACLE_HOME=IGD_ORACLE_HOME
   export WL_HOME=IGD_ORACLE_HOME/wlserver
   chmod 750 _OIGOAMIntegration.sh OIGOAMIntegration.sh
   ./OIGOAMIntegration.sh -configureLDAPConnector
   ```

Add Missing Object Classes

If any users existed in LDAP prior to enabling the Oracle Identity Manager, then these new users may be missing the object classes used to control OIM/OAM integration. To add these missing object classes to these users, run the following commands:

> **Note:**
>
> To successfully execute this process, the `ldapsearch` binary is required to be in your user's PATH and the `screen` package is required to be installed on your host.

1. Change directory to *IGD_ORACLE_HOME*/idm/server/ssointg/config
2. Edit the file `addMissingObjectClasses.config` updating the properties as shown below:

   ```
   IDSTORE_DIRECTORYTYPE: OUD
   IDSTORE_HOST: idstore.example.com
   IDSTORE_PORT: 1389
   IDSTORE_BINDDN: cn=oudadmin
   IDSTORE_USERSEARCHBASE: cn=Users,dc=example,dc=com
   ```

   Save the file when done.

   Table 17-5 Properties of `addMissingObjectClasses.config`

Attribute	Description
IDSTORE_HOST	It is the Load Balancer name for the LDAP directory. For example; **idstore.example.com**
IDSTORE_PORT	It is the LDAP port on the load balancer. For example; 1389 for OUD.
IDSTORE_DIRECTORYTYPE	It is the type of LDAP directory you are using (OUD).
IDSTORE_BINDDN	It is the credential used to connect to the directory to perform administrative actions, for example, **oudadmin** for OUD.
IDSTORE_USERSEARCHBASE	It is the location in the directory where user information is stored.

3. Execute the script OIGOAMIntegration.

4. For example:

   ```
   cd IGD_ORACLE_HOME/idm/server/ssointg/bin
   export JAVA_HOME=JAVA_HOME
   export ORACLE_HOME=IGD_ORACLE_HOME
   export WL_HOME=IGD_ORACLE_HOME/wlserver
   ./OIGOAMIntegration.sh -addMissingObjectClasses
   ```

   You will be prompted to enter the password of the LDAP directory administrator account.

Restart Domains

Restart the **IAMAccessDomain** and the **IAMGovernanceDomain** domains.

Integrating Oracle Identity Governance and Oracle Access Manager

You have to complete several tasks to integrate Oracle Identity Governance and Oracle Access Manager. These tasks include creating the WLS authentication providers, deleting OIMSignatureAuthenticator and recreating OUDAuthenticator, adding the administration role to the new administration group, and so on.

- Configuring SSO Integration in the IAMGovernanceDomain
- Enable OAM Notifications
- Update Value of MatchLDAPAttribute in oam-config.xml
- Update TapEndpoint URL

Configuring SSO Integration in the IAMGovernanceDomain

Having deployed the connector the next step in the process is the configuration of SSO in the domain. In order to do this you need to perform the following steps:

1. Change directory to *IGD_ORACLE_HOME/idm/server/ssointg/config*
2. Edit the file `configureSSOIntegration.config` updating the properties in the section **configureSSOIntegration** as shown below:

   ```
   ##-----------------------------------------------------------##
   ## [configureSSOIntegration]
   OAM_HOST: login.example.com
   OAM_PORT: 443
   OAM_PORT: 80
   ACCESS_SERVER_HOST:OAMHOST1.example.com
   ACCESS_SERVER_PORT: 5575
   OAM_SERVER_VERSION: 12c
   WEBGATE_TYPE: ohsWebgate12c
   COOKIE_DOMAIN: example.com
   OAM_TRANSFER_MODE: open
   OIM_LOGINATTRIBUTE: uid
   SSO_INTEGRATION_MODE: CQR
   OAM11G_WLS_ADMIN_HOST: IADADMINVHN.example.com
   OAM11G_WLS_ADMIN_PORT: 7001
   OAM11G_WLS_ADMIN_USER: weblogic
   ```

```
OAM11G_WLS_ADMIN_PASSWD: <PASSWORD>
OAM11G_IDSTORE_NAME: OAMIDSTORE
## Required if OAM_TRANSFER_MODE is not OPEN
OIM_WLSHOST:IGDADMINVHN.example.com
OIM_WLSPORT: 7101
OIM_WLSADMIN: weblogic
IDSTORE_OAMADMINUSER_PWD: <password>
OIM_SERVER_NAME: WLS_OIM1
IDSTORE_OAMADMINUSER: oamadmin
```

Save the file when done.

Where:

Table 17-6 Configure SSOIntegration Properties

Attribute	Description
OAM_HOST	It is the listen address of the front end load balancer for the OAM cluster.
OAM_PORT	It is the port of the front end load balancer for the OAM cluster.
ACCESS_SERVER_HOST	It is always the same as the OAM_HOST.
ACCESS_SERVER_PORT	It is the port number for `OAM PROXY PORT`.
ACCESS_GATE_ID	It is the name of the WebGate agent created in Creating a Configuration File.
COOKIE_DOMAIN	It is the value assigned in Creating a Configuration File.
OAM_TRANSFER_MODE	It is the value assigned in Creating a Configuration File.
OIM_LOGINATTRIBUTE	It is the LDAP field containing the users login attribute usually `uid` or `cn`.
OAM11G_WLS_ADMIN_HOST	It is the listen address of the Administration Server in the domain `IAMAccessDomain`. For example: `IADADMINVHN`.
OAM11G_WLS_ADMIN_PORT	It is the listen port of the Administration Server in the domain `IAMAccessDomain`. For example: `7001`.
OAM11G_WLS_ADMIN_PASSWD	Optional password for *OAM11G_WLS_ADMIN_USER*.
OAM11G_WLS_ADMIN_USER	It is the Administration User of the IAD Administration Server.
OIM_WLSHOST	The listen address of the OIM Administration server for example `IGDADMINVHN.example.com`
OIM_WLSPORT	The listen port of the OIM Administration Server. For example: `7101`.
OIM_WLSADMIN	The administration user of the OIM Administration Server. For example: `weblogic`.
OIM_SERVER_NAME	It is the name of the OIM Managed Server that is running. For example: `WLS_OIM1`.

Table 17-6 (Cont.) Configure SSOIntegration Properties

Attribute	Description
IDSTORE_OAMADMINUSER	The value assigned to *IDSTORE_OAMADMINUSER* in Creating a Configuration File.
IDSTORE_OAMADMINUSER_PWD	It is optional. It contains the password of the *IDSTORE_OAMADMINUSER* account.
OAM_SERVER_VERSION	It is the version of OAM used for the integration.
WEBGATE_TYPE	It is the type of WebGate used for the integration.
OAM11G_IDSTORE_NAME	The name of the IDStore configured in OAM, the default name is `OAMIDSTORE`.

3. Execute the script OIGOAMIntegration for configuring SSO Integration.

   For example:

   ```
   cd IGD_ORACLE_HOME/idm/servers/ssointg/bin
   export JAVA_HOME=JAVA_HOME
   export ORACLE_HOME=IGD_ORACLE_HOME
   export WL_HOME=IGD_ORACLE_HOME/wlserver
   ./OIGOAMIntegration.sh -configureSSOIntegration
   ```

4. Restart the domains **IAMAccessDomain** and **IAMGovernanceDomain**.

Enable OAM Notifications

Having deployed the connector the next step in the process is to tell OIM how to interact with OAM for terminating a user session after a user has been expired or terminated. In order to do this you need to perform the following steps:

1. Change directory to *IGD_ORACLE_HOME*/idm/server/ssointg/config.
2. Edit the file `enableOAMSessionDeletion.config` updating the properties in the section **enableOAMNotifications** as shown below:

   ```
   ##-----------------------------------------------------------##

   ## [enableOAMNotifications]
   OIM_WLSHOST: IGDADMINVHN.example.com
   OIM_WLSPORT: 7101
   OIM_WLSADMIN: weblogic
   IDSTORE_DIRECTORYTYPE: OUD
   IDSTORE_HOST: idstore.example.com
   IDSTORE_PORT: 1389
   IDSTORE_BINDDN: cn=oudadmin
   IDSTORE_GROUPSEARCHBASE: cn=Groups,dc=example,dc=com
   IDSTORE_SYSTEMIDBASE: cn=systemids,dc=example,dc=com
   IDSTORE_OAMADMINUSER: oamAdmin
   IDSTORE_OAMSOFTWAREUSER: oamLDAP
   IDSTORE_USERSEARCHBASE: cn=Users,dc=example,dc=com
   OIM_SERVER_NAME: WLS_OIM1
   ```

   Where:

Table 17-7 Properties of enableOAMSessionDeletion

Attribute	Description
OIM_WLSHOST	It is the listen address of the Administration Server in the domain `IAMGovernanceDomain`. For example: `IGDADMINVHN.example.com`.
OIM_WLSPORT	It is the port of the Administration Server in the domain `IAMGovernanceDomain`. For example: `7101`.
OIM_WLSADMIN	It is the name of the WebLogic administrator in the `IAMGovernanceDomain`. For example: `weblogic`.
IDSTORE_HOST	It is the load balancer name for the LDAP directory. For example: `idstore.example.com`.
IDSTORE_PORT	It is the LDAP port of the load balancer. For example: `1389` for OUD.
IDSTORE_BINDDN	It is the credential used to connect to the directory to perform administrative actions. For example: `oudadmin` for OUD.
IDSTORE_GROUPSEARCHBASE	It is the location in the directory where Groups are Stored.
IDSTORE_SYSTEMIDBASE	It is the location of a container in the directory where system users can be placed when you do not want them in the main user container.
IDSTORE_OAMADMINUSER	It is the name of the user you want to create as your Access Manager Administrator.
IDSTORE_OAMSOFTWAREUSER	A user that gets created in LDAP that is used when Access Manager is running to connect to the LDAP server.
IDSTORE_USERSEARCHBASE	It is the location in the directory where users are stored.
OIM_SERVER_NAME	The name of the OIM server. For example: `oim_server1`.

3. Execute the script `OIGOAMIntegration` for enabling notifications.

   For example:

   ```
   cd IGD_ORACLE_HOME/idm/servers/sointg/bin
   export JAVA_HOME=JAVA_HOME
   export ORACLE_HOME=IGD_ORACLE_HOME
   export WL_HOME=IGD_ORACLE_HOME/wlserver
   ./OIGOAMIntegration.sh -enableOAMSessionDeletion
   ```

Update Value of MatchLDAPAttribute in oam-config.xml

To complete the Oracle Identity Governance integration with Oracle Access Manager, one of the settings in the Oracle Access Manager's `oam-config.xml` file needs to be changed. As of version 12c, this file is stored in the database and should not be edited directly.

The procedure below shows how to use the REST API to change one of the values in the `oam-config.xml` file:

> **Note:**
>
> Ensure that the cURL package has been added to the host by executing `which curl` at the command line. If the package is not installed, an administrator must install the package by executing `yum install curl`.

1. Find the component number of the DAPModule, by executing the following:

   ```
   curl -i -u weblogic:<password> http://IADADMINVHN:7001/iam/admin/config/api/v1/
   config?path=/DeployedComponent/Server/NGAMServer/Profile/AuthenticationModules/
   DAPModules
   ```

   where:

   - `weblogic`: The WebLogic administrative user configured in OUD.
   - <`password`>: The above user's password.
   - `IADADMINVHN`: The VIP at which the Access Manager domain Admin Console runs.
   - `7001`: The port at which the Access Manager domain Admin Console runs.

   Example output:

   ```
   HTTP/1.1 200 OK
   Date: Tue, 09 Jul 2019 20:30:33 GMT
   Content-Length: 625
   Content-Type: text/xml
   X-ORACLE-DMS-ECID: 6f9baf65-751b-4fc9-b2e1-ade5b38063ff-00000427
   X-ORACLE-DMS-RID: 0
   Set-Cookie: JSESSIONID=g3LYbkLA2bs5-9zfoMBqKTBbk0mky_8URGgzFnbNkm8n3tK63tq4!
   1064195705; path=/; HttpOnly

   <Configuration xmlns="http://www.w3.org/2001/XMLSchema" schemaLocation="http://
   higgins.eclipse.org/sts/Configuration Configuration.xsd" Path="/DeployedComponent/
   Server/NGAMServer/Profile/AuthenticationModules/DAPModules">

   <Setting Name="DAPModules" Type="htf:map">
       <Setting Name="7DASE52D" Type="htf:map">
         <Setting Name="MAPPERCLASS"
   Type="xsd:string">oracle.security.am.engine.authn.internal.executor.DAPAttributeMappe
   r</Setting>
         <Setting Name="MatchLDAPAttribute" Type="xsd:string”>User Name</Setting>
         <Setting Name="name" Type="xsd:string">DAP</Setting>
       </Setting>
     </Setting>
   ```

   > **Note:**
   >
   > The component number under the line that reads: "<Setting Name="DAPModules" Type="htf:map">" This will need to be used for the configuration change. In the above example, "7DASE52D" is the component number. The value which will need to be changed is the value of MatchLDAPAttribute. In the above example, "User Name" is the current value.

2. Change directory to `/tmp` and create a configuration file `MatchLDAPAttribute_input.xml` with the following contents:

```
<Configuration>
  <Setting Name="MatchLDAPAttribute" Type="xsd:string" Path="/DeployedComponent/
Server/NGAMServer/Profile/AuthenticationModules/DAPModules/7DASE52D/
MatchLDAPAttribute">uid</Setting>
</Configuration>
```

> **Note:**
>
> The component number noted from above is inserted between `DAPModules` and the `MatchLDAPAttribute` portions of the path. The configuration file will change the value of `MatchLDAPAttribute` from *User Name* to *uid*.

3. Insert the change back into the OAM configuration, by executing the following:

   ```
   curl -u weblogic:<password> -H 'Content-Type: text/xml' -X PUT http://
   IAMADMINVHN:7001/iam/admin/config/api/v1/config -d @MatchLDAPAttribute_input.xml
   ```

4. Validate the change with the same command you originally used to query the component, noting the value of the `MatchLDAPAttribute` tag:

   ```
   curl -i -u weblogic:<password> http://IADADMINVHN:7001/iam/admin/config/api/v1/
   config?path=/DeployedComponent/Server/NGAMServer/Profile/AuthenticationModules/
   DAPModules
   ```

   Example output:

   ```
   HTTP/1.1 200 OK
   Date: Tue, 09 Jul 2019 20:30:33 GMT
   Content-Length: 625
   Content-Type: text/xml
   X-ORACLE-DMS-ECID: 6f9baf65-751b-4fc9-b2e1-ade5b38063ff-00000427
   X-ORACLE-DMS-RID: 0
   Set-Cookie: JSESSIONID=g3LYbkLA2bs5-9zfoMBqKTBbk0mky_8URGgzFnbNkm8n3tK63tq4!
   1064195705; path=/; HttpOnly

   <Configuration xmlns="http://www.w3.org/2001/XMLSchema" schemaLocation="http://
   higgins.eclipse.org/sts/Configuration Configuration.xsd" Path="/DeployedComponent/
   Server/NGAMServer/Profile/AuthenticationModules/DAPModules">

   <Setting Name="DAPModules" Type="htf:map">
       <Setting Name="7DASE52D" Type="htf:map">
         <Setting Name="MAPPERCLASS"
   Type="xsd:string">oracle.security.am.engine.authn.internal.executor.DAPAttributeMappe
   r</Setting>
         <Setting Name="MatchLDAPAttribute" Type="xsd:string”>uid</Setting>
         <Setting Name="name" Type="xsd:string">DAP</Setting>
       </Setting>
     </Setting>
   ```

Update TapEndpoint URL

For OAM/OIM integration to work you must update the OAM TapEndpoint URL you do this by performing the following steps.

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

   ```
   http://igdadmin.example.com/em
   ```

   OR

```
http://IGDADMINVHN.example.com:7101/em
```

The Administration Server host and port number were in the URL on the End of Configuration screen (Writing Down Your Domain Home and Administration Server URL). The default Administration Server port number is 7101.

2. Click **WebLogic Domain**, and click **System MBean Browser**.

   In the search box, enter **SSOIntegrationMXBean**, and click **Search**. The mbean is displayed.

3. Set the value of **TapEndpointURL** to

   ```
   https://login.example.com/oam/server/dap/cred_submit
   ```

4. Click **Apply**.

Running the Reconciliation Jobs

Run the Oracle Identity Governance domain to import the LDAP user names into the Oracle Identity Governance database.

To run the reconciliation jobs:

1. Log in to the OIM System Administration Console as the user `xelsysadm`.
2. Click **Scheduler** under **System Configuration**.
3. Enter `SSO*` in the search box.
4. Click the arrow for the **Search Scheduled Jobs** to list all the schedulers.
5. Select **SSO User Full Reconciliation**.
6. Click **Run Now** to run the job.
7. Repeat for **SSO Group Create And Update Full Reconciliation**.
8. Log in to the OIM System Administration Console and verify that the user `weblogic_iam` is visible.

Update the SOA Integration URL

Oracle Identity Manager connects to SOA as SOA administrator, with the username `weblogic`.

Perform the following post installation steps to enable Oracle Identity Manager to work with the Oracle WebLogic Server administrator user. This enables Oracle Identity Manager to connect to SOA:

> **Note:**
>
> For the SOAConfig Mbean to be visible, at least one OIM Managed Server must be running.

1. Log in to Enterprise Manager Fusion Middleware Control of the IAMGovernanceDomain, as the `weblogic` user
2. Click **WebLogic Domain**, and click **System MBean Browser**.
3. Select **Search**, enter `SOAConfig`, and click **Search**.

4. Ensure that the username is set to `weblogic`.
5. Update the SOAP URL to the following:

   ```
   http://igdinternal.example.com:7777/
   ```

6. Update the SOA Config RMI URL to the following:

   ```
   http://igdinternal.example.com:7777/
   ```

7. Click **Apply**.

Configuring OIM Workflow Notifications to be Sent by Email

OIM uses the human workflow, which is integrated with the SOA workflow. The SOA server configures email to receive the notifications that are delivered to the user mailbox. The user can accept or reject the notifications.

Both incoming and outgoing email addresses and mailboxes dedicated to the portal workflow are required for the full functionality. See Configuring Human Workflow Notification Properties in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

To configure the OIM workflow notifications:

1. Log in to the Fusion Middleware Control by using the administrators account. For example, `weblogic_iam`.
2. Expand the Target Navigation panel and navigate to **SOA** > **soa-infra (soa_server1)** service.
3. From the SOA infrastructure drop-down, select **SOA Administration** > **Workflow Properties**.
4. Set the Notification mode to **Email**. Provide the correct e-mail address for the notification service.
5. Click **Apply** and confirm when prompted.
6. Verify the changes.
7. Expand **Target Navigation**, select **User Messaging Service**, and then **usermessagingdriver-email (soa_servern)**. Each SOA Managed Server that is running will have a driver. Only one of these entries should be selected.
8. From the **User Messaging Email Driver** drop-down list, select **Email Driver Properties**.
9. Click **Create** if the email driver does not exist already.
10. Click **Test** and verify the changes.
11. Click **OK** to save the email driver configuration.
12. Restart the SOA cluster. No configuration or restart is required for OIM.

Adding the wsm-pm Role to the Administrators Group

After you configure a new LDAP-based Authorization Provider and restart the Administration Server, add the enterprise deployment administration LDAP group (OIMAdministrators) as a member to the `policy.Updater` role in the `wsm-pm` application stripe.

1. Sign in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_iam`.
2. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.
3. Select the **wsm-pm** application stripe from the Application Stripe drop-down menu.
4. Click the triangular icon next to the role name text box to search for all role names in the wsm-pm application stripe.
5. Select the row for the **policy.Updater** role to be edited.
6. Click the Application Role **Edit** icon to edit the role.
7. Click the Application Role **Add** icon on the Edit Application Role page.
8. In the Add Principal dialog box, select **Group** from the **Type** drop-down menu.
9. To search for the enterprise deployment administrators group, enter the group name `WLSAdministrators` in the **Principal Name Starts With** field and click the right arrow to start the search.
10. Select the appropriate administrators group in the search results and click **OK**.
11. Click **OK** on the Edit Application Role page.

Adding the Oracle Access Manager Load Balancer Certificate to the Oracle Keystore Service

The Oracle Identity Governance to Business Intelligence Reports link inside of the Self Service application requires that the SSL certificate used by the load balancer be added to the Oracle Keystore Service Trusted Certificates.

To add the certificate, do the following:

1. Create a directory to hold user created keystores and certificates.

   For example:

   ```
   mkdir SHARED_CONFIG_DIR/keystores
   ```

2. Obtain the certificate from the load balancer. You can obtain the load balancer certificate from using a browser, such as Firefox. However, the easiest way to obtain the certificate is to use the `openssl` command. The syntax of the command is as follows:

   ```
   openssl s_client -connect LOADBALANCER -showcerts </dev/null 2>/dev/null|
   openssl x509 -outform PEM>SHARED_CONFIG_DIR/keystores/LOADBALANCER.pem
   ```

   For example:

   ```
   openssl s_client -connect login.example.com:443 -showcerts </dev/null
   2>/dev/null|openssl x509 -outform PEM >SHARED_CONFIG_DIR/keystores/
   login.example.com.pem
   ```

The `openssl` command saves the certificate to a file called `login.example.com.pem` in *SHARED_CONFIG_DIR*/`keystores`.

3. Load the certificate into the Oracle Keystore Service using WLST.
   a. Connect to WLST using the following command:

      ```
      ORACLE_HOME/oracle_common/common/bin/wlst.sh
      ```

   b. Connect to the Administration Server using the following command:

      ```
      connect('<AdminUser>','<AdminPwd>','t3://<Adminserverhost>:<Adminserver
      port>')
      ```

   c. Load the certificate using the following commands:

      ```
      svc = getOpssService(name='KeyStoreService')
      svc.importKeyStoreCertificate(appStripe='system',name='trust',password='
      ', keypassword='',alias='<CertificateName>',type='TrustedCertificate',
      filepath='/<SHARED_CONFIG_DIR>/keystores/<LOADBALANCER>.pem')
      ```

   d. Synchronize the Keystore Service with the file system using the following command:

      ```
      syncKeyStores(appStripe='system', keystoreFormat='KSS')
      ```

      For example:

      ```
      connect('weblogic','password','t3://IGDADMINVHN.example.coml:7101')
      svc = getOpssService(name='KeyStoreService')
      svc.importKeyStoreCertificate(appStripe='system',name='trust',password='
      ', keypassword='',alias='login.example.com',type='TrustedCertificate',
      filepath='/u01/oracle/config/keystores/login.example.com.pem')
      syncKeyStores(appStripe='system',keystoreFormat='KSS')
      exit()
      ```

You will need to restart the domain for the changes to take effect. The default password for the Node Manager keystores is *COMMON_IAM_PASSWORD*. You will be prompted to confirm that the certificate is valid.

Restarting the IAMGovernanceDomain

For the above changes to take effect, you must restart the domain.

1. Shut down the Managed Servers WLS_OIM1 and WLS_OIM2.
2. Shut down the Managed Servers WLS_SOA1 and WLS_SOA2.
3. Shut down the Managed Servers WLS_WSM1 and WLS_WSM2.
4. Shut down the Administration Server.
5. Restart the Administration Server.
6. Start the Managed Servers WLS_SOA1 and WLS_SOA2.
7. Start the Managed Servers WLS_OIM1 and WLS_OIM2.
8. Start the Managed Servers WLS_WSM1 and WLS_WSM2.

If you have performed the workaround as described in the Update Value of MatchLDAPAttribute in oam-config.xml, then you must also restart the OAM domain.

Shut down and restart the Administration Server and all the Managed Servers (WLS_AMA1, WLS_AMA2, WLS_OAM1, WLS_OAM2).

Setting Challenge Questions

If you have integrated OAM and OIM, then after the environment is ready, you need to set up the challenge questions for your system users.

To set up the challenge questions, log in to Identity Self Service using the URL: https://prov.example.com/identity.

Log in with your user name and when prompted, add the challenge questions. You should set up these questions for the following users:

- `xelsysadm`
- `weblogic_iam`
- `oamadmin`

Integrating Oracle Identity Manager with Oracle Business Intelligence Publisher

Oracle Identity Manager comes with a number of prebuilt reports that can be used to provide information about Oracle Identity and Access Management.

Oracle Identity Manager reports are classified based on the functional areas such as Access Policy Reports, Request and Approval Reports, Password Reports, and so on. It is no longer named Operational and Historical. These reports are not generated through Oracle Identity Manager but by the Oracle Business Intelligence Publisher (BIP). Oracle Identity Manager reports provide a restriction for Oracle BI Publisher.

The setup of a highly available enterprise deployment of Oracle BI Publisher is beyond the scope of this document. For more information, see Understanding the Business Intelligence Enterprise Deployment Topology in the *Enterprise Deployment Guide for Business Intelligence*.

> **Note:**
>
> During BI configuration for Oracle Identity Manager, you must configure only Business Intelligence Publisher. If you select other components during BI Publisher configuration, such as Business Intelligence Enterprise Edition and Essbase, the integration with Oracle Identity Manager may not work. See Configuring Reports in *Developing and Customizing Applications for Oracle Identity Manager*

- Creating a User to Run BI Reports
- Configuring Oracle Identity Manager to Use BI Publisher
  You can set up Oracle BI Publisher to generate Oracle Identity Manager reports.
- Assigning the BIServiceAdministrator Role to idm_report
- Storing the BI Credentials in Oracle Identity Governance

- Creating OIM and BPEL Data Sources in BIP
- Deploying Oracle Identity Governance Reports to BI
- Enable Certification Reports
- Validating the Reports

Creating a User to Run BI Reports

You may ignore this section if you already have a user to run reports in your Business Intelligence domain.

If you need to create a user in your BI Publisher domain to run reports, use the following `LDIF` command to create a user in the LDAP directory.

1. Create a file called `report_user.ldif` with the following contents:

   ```
   dn: cn=idm_report,cn=Users,dc=example,dc=com
   changetype: add
   orclsamaccountname: idm_report
   givenname: idm_report
   sn: idm_report
   userpassword: <password>
   mail: idm_report
   objectclass: top
   objectclass: person
   objectclass: organizationalPerson
   objectclass: inetorgperson
   objectclass: orcluser
   objectclass: orcluserV2
   uid: idm_report
   cn: idm_report
   ```

2. Save the file.
3. Load the file into the LDAP directory using the following command:

   ```
   ldapmodify -D cn=oudadmin -h idstore.example.com -p 1389 report_user.ldif
   ```

Configuring Oracle Identity Manager to Use BI Publisher

You can set up Oracle BI Publisher to generate Oracle Identity Manager reports.

To configure Oracle Identity Manager to use the BI Publisher:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control using the URL:

   `http://igdadmin.example.com/em`

2. Click WebLogic Domain, and then select **System MBean Browser**.
3. Enter `XMLConfig.DiscoveryConfig` as the search criteria and click **Search**.

   The XMLConfig.DiscoveryConfig MBean is displayed.

4. Update the value of the **Discovery Config BI publisher URL** to the BIP URL. For example, `http://bi.example.com`
5. Click **Apply**.

Assigning the BIServiceAdministrator Role to idm_report

If you are using LDAP as your identity store in the Business Intelligence (BI) domain, you must have created an LDAP authenticator in the BI domain. You can view the user and group names stored within LDAP.

The Oracle Identity Manager (OIM) system administration account (for example, `idm_report`) needs to be assigned the `BIServiceAdministrator` role, to generate reports.

To assign this role:

1. Ensure that the OIM administrator user is visible in the domain by logging in to the BI publisher WebLogic Console using the following URL:

   ```
   http://biadmin.example.com/console
   ```

2. Click **Security Realms**, and then click **myrealm**.
3. Go to the **Users and Groups** tab.
4. Look at the list of users and ensure that the user OIM Administration User (`idm_report`) is in the list of users.
5. Sign in to the BI Fusion Middleware Control by using the URL `http://biadmin.example.com/em` and the administrator's account. For example: *weblogic_bi*.
6. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.
7. From the **Application Stripe** drop-down list, select **obi**.
8. Click the triangular icon next to the role name text box to search for all role names in the **obi** application stripe.
9. Select the row for the **BIServiceAdministrator** role to edit.
10. Click the Application Role Edit icon to edit the role.
11. Click the Application Role Add icon on the Edit Application Role page.
12. In the Add Principal dialog box, select **User** from the **Type** drop-down menu.
13. To search for the `idm_report` user, enter the user name `idm_report` in the **Principal Name Starts With** field and click the right arrow to start the search.
14. Select the appropriate user in the search results and click **OK**.
15. Click **OK** on the Edit Application Role page.

Storing the BI Credentials in Oracle Identity Governance

To configure BIP credentials in Oracle Identity Manager:

1. Log in to the Oracle Enterprise Manager using the url

   ```
   http://igdadmin.example.com/em
   ```

2. In the left pane, expand the **Weblogic Domain**. The domain name is displayed.
3. Right-click the domain name, and navigate to **Security**, and then **Credentials**. A list of maps in the credential store, including the oim map, is displayed.
4. Expand the oim map. A list of entries of type Password is displayed.
5. Edit the `BIPWSKey` key if it already exists, or create a new one with the following values:

Table 17-8 Properties of a new CSF entry

Attribute	Value
Select Map	oim
Key	BIPWSKey
Type	Password
Username	`idm_report`
Password	`idm_report` password
Description	Login credentials for BI Publisher web service

Creating OIM and BPEL Data Sources in BIP

Create OIM Datasource

Oracle BIP must be connected to the OIM and SOA database schemas to run a report.

In order to do this you need to create BIP datasources using the following procedure:

1. Login to the BI Publisher Home page using the URL `https://bi.example.com/xmlpserver`
2. Click the **Administration** link on the top of the BI Publisher Home page. The BI Publisher Administration page is displayed.
3. Under Data Sources, click **JDBC Connection** link. The Data Sources page is displayed.
4. In the JDBC tab, click **Add Data Source** to create a JDBC connection to your database. The Add Data Source page is displayed.
5. Enter values in the following fields:

Table 17-9 OIM Add Data Source Attributes

Attributes	Value
Data Source Name	Specify the Oracle Identity Governance JDBC connection name. For example, OIM JDBC.
Driver Type	Select **Oracle 11g** for an 11g database and **Oracle 12c** for a 12c database
Database Driver Class	Specify a driver class to suit your database, such as `oracle.jdbc.OracleDriver`
Connection String	Specify the database connection details in the format `jdbc:oracle:thin:@HOST_NAME:PORT_NUMBER/SID`. For example, `jdbc:oracle:thin:@igddbscan:1521/oim.example.com`
User name	Specify the Oracle Identity Governance database user name for example IGD_OIM
Password	Specify the Oracle Identity Governance database user password.

6. Click **Test Connection** to verify the connection.

7. Click **Apply** to establish the connection.
8. If the connection to the database is established, a confirmation message is displayed indicating the success.
9. Click **Apply**.

In the JDBC page, you can see the newly defined Oracle Identity Governance JDBC connection in the list of JDBC data sources.

Create BPEL Datasource

1. Login to the BI Publisher Home page using the URL `https://bi.example.com/xmlpserver`.
2. Click the **Administration** link on the **BI Publisher** home page. The **BI Publisher Administration** page is displayed.
3. Under Data Sources, click **JDBC Connection** link. The Data Sources page is displayed.
4. In the JDBC tab, click **Add Data Source** to create a JDBC connection to your database. The Add Data Source page is displayed.
5. Enter values in the following fields:

Table 17-10 JDBC Add Data Source Attributes

Attributes	Value
Data Source Name	Specify the Oracle Identity Governance JDBC connection name. For example, BPEL JDBC.
Driver Type	Oracle 12c
Database Driver Class	Specify a driver class to suit your database, such as `oracle.jdbc.OracleDriver`
Connection String	Specify the database connection details in the format `jdbc:oracle:thin:@HOST_NAME:PORT_NUMBER/SID`. For example, `jdbc:oracle:thin:@igddbscan:1521/oim.example.com`
User name	Specify the Oracle Identity Governance database user name for example IGD_SOAINFRA.
Password	Specify the Oracle Identity Governance database user password.

6. Click **Test Connection** to verify the connection.
7. Click **Apply** to establish the connection.
8. If the connection to the database is established, a confirmation message is displayed indicating the success.
9. Click **Apply**.

In the JDBC page, you can see the newly defined Oracle Identity Governance JDBC connection in the list of JDBC data sources.

Deploying Oracle Identity Governance Reports to BI

After **BI Publisher** is integrated with Oracle Identity Governance, you can deploy the predefined reports for using them. To deploy Oracle Identity Manager reports:

1. Copy and unzip the predefined report `IGD_ORACLE_HOME/idm/server/reports/oim_product_BIPReports_12c.zip` located on `OIMHOST1` file to the directory `Shared_Storage_location/biconfig/bidata`.

   > **Note:**
   >
   > The *Shared_Storage_Location* is defined in the `ASERVER_HOME/config/fmwconfig/bienv/core/bi-environment.xml` file.

2. Add folder level permission to the **BIServiceAdministrator** BI application role to view and run the predefined Oracle Identity Governance reports. To do so:
   - Login to Oracle BI Publisher `https://bi.example.com/xmlpserver` by using the WebLogic admin credentials.
   - Click the **Catalog** link at the top. The Oracle Identity Manager named folder under shared folders is displayed in the left pane. Select the Oracle Identity Manager named folder.
   - Click **Permissions** option under the **Tasks** window on the bottom left.
   - Click the plus sign and perform a blank search on the available role.
   - Select the **BI Service Administrator** role, and add to the right panel.
   - Click **Ok**.
3. Logout as WebLogic user.
4. Login as the Oracle Identity Manager system administrator user to **BI Publisher console**.
5. Run the Oracle Identity Manager reports.

Enable Certification Reports

Select or deselect the **Enable Certification Reports** option to enable or disable the certification reports. To enable the generation of certification reports, after configuring the **BI Publisher** credentials and URL, perform the following:

1. Log in to the Oracle Identity Self Service using the url: `https://prov.example.com/identity`.
2. Click the **Compliance** tab.
3. Click the **Identity Certification** box.
4. Select **Certification Configuration**. The Certification Configuration page is displayed.
5. Select the **Enable Certification Reports**.
6. Click **Save**.

> **Note:**
>
> By default, the **Compliance** tab is not shown. If you want to enable compliance functionality, you must fist set the `OIGIsIdentityAuditorEnabled` property to `true` in the Sysadmin Console (located in the **Configuration Properties** section).

Validating the Reports

We need to create the sample data source to generate reports against the sample data source.

Creating the Sample Reports

To view an example report data without running a report against the production JDBC Data Source, generate a sample report against the sample data source. Create the sample data source before you can generate the sample reports.

- Generating Reports Against the Sample Data Source
- Generating Reports Against the Oracle Identity Manager JDBC Data Source
- Generating Reports Against the BPEL-Based JDBC Data Source

Generating Reports Against the Sample Data Source

After you create the sample data source, you can generate sample reports against it by performing the following steps:

1. Login to Oracle BI Publisher using the url : `https://bi.example.com/xmlpserver`.
2. Click **Shared Folders**.
3. Click **Oracle Identity Manager Reports**.
4. Select **Sample Reports**.
5. Click **View** for the sample report you want to generate.
6. Select an output format for the sample report and click **View**.

The sample report is generated.

Generating Reports Against the Oracle Identity Manager JDBC Data Source

To generate reports against the OIM JDBC data source, navigate to the Oracle Identity Manager reports by logging in to the Oracle BI Publisher, and select an output format for the report you want to generate.

To generate reports against the Oracle Identity Manager JDBC data source:

1. Log in to Oracle BI Publisher using the url :`https://bi.example.com/xmlpserver`.
2. Navigate to Oracle Identity Manager reports. To do so:
   - In the **BI Publisher** home page, under Browse or Manage, click **Catalog Folders**. Alternatively, you can click **Catalog** at the top of the page.

     The Catalog page is displayed with a tree structure on the left side of the page and the details on the right.
   - On the left pane, expand **Shared Folders**, and navigate to the Oracle Identity Manager. All the objects in the Oracle Identity Manager folder are displayed.

You are ready to navigate to BI Publisher 12*c* and use the Oracle Identity Manager BI Publisher reports.

3. Click **View** under the report you want to generate.
4. Select an output format for the report and click **View**.

The report is generated.

Generating Reports Against the BPEL-Based JDBC Data Source

Some reports have a secondary data source, which is BPEL-based JDBC data source. This section describes how to generate reports against the BPEL-based JDBC data source.

Reports With Secondary Data Source

The following four reports have a secondary data source, which connects to the BPEL database to retrieve the BPEL data:

- Task Assignment History
- Request Details
- Request Summary
- Approval Activity

These reports have a secondary data source (BPEL-based JDBC data source) called BPEL JDBC. To generate reports against the BPEL-based JDBC data source:

1. Log in to Oracle BI Publisher using the url: `https://bi.example.com/xmlpserver`.
2. Navigate to the Oracle Identity Manager reports. To do so:
   - In the **BI Publisher** home page, under Browse or Manage, click **Catalog Folders**. Alternatively, you can click **Catalog** at the top of the page.

     The catalog page is displayed with a tree structure on the left side of the page and the details on the right.
   - On the left pane, expand **Shared Folders**, and navigate to the Oracle Identity Manager. All the objects in the Oracle Identity Manager folder is displayed.

     Navigate to the BI Publisher 12*c* and use the Oracle Identity Manager BI Publisher reports.
3. Select the report you want to generate and click **Open**.
4. Select an output format for the report, and click **Apply**.

The report is generated based on the BPEL-based JDBC data source.

- Adding the Business Intelligence Load Balancer Certificate to Oracle Keystore Trust Service

Adding the Business Intelligence Load Balancer Certificate to Oracle Keystore Trust Service

The Oracle Identity Governance to Business Intelligence Reports link inside of the Self Service application requires that the SSL certificate used by the load balancer be added to the Oracle Keystore Service Trusted Certificates.

To add the certificate:

1. Create a directory to hold user created keystores and certificates.

For example:

```
mkdir SHARED_CONFIG_DIR/keystores
```

2. Obtain the certificate from the load balancer. You can obtain the load balancer certificate from using a browser, such as Firefox. However, the easiest way to obtain the certificate is to use the `openssl` command. The syntax of the command is as follows:

   ```
   openssl s_client -connect LOADBALANCER -showcerts </dev/null 2>/dev/null|
   openssl x509 -outform PEM>SHARED_CONFIG_DIR/keystores/LOADBALANCER.pem
   ```

   For example:

   ```
   openssl s_client -connect bi.example.com:443 -showcerts </dev/null 2>/dev/
   null|openssl x509 -outform PEM>SHARED_CONFIG_DIR/keystores/
   bi.example.com.pem
   ```

   The `openssl` command saves the certificate to a file called `bi.example.com.pem` in `SHARED_CONFIG_DIR/keystores`.

3. Load the certificate into the Oracle Keystore Service using WLST.

   a. Connect to WLST using the following command:

      ```
      ORACLE_HOME/oracle_common/common/bin/wlst.sh
      ```

   b. Connect to the Administration Server using the following command:

      ```
      connect('<AdminUser>','<AdminPwd>','t3://<Adminserverhost>:<Adminserver
      port>')
      ```

   c. Load the certificate using the following commands:

      ```
      svc = getOpssService(name='KeyStoreService')
      svc.importKeyStoreCertificate(appStripe='system',name='trust',password='
      ', keypassword='',alias='<CertificateName>',type='TrustedCertificate',
      filepath='/<SHARED_CONFIG_DIR>/keystores/<LOADBALANCER>.pem')
      ```

   d. Synchronize the Keystore Service with the file system using the following command:

      ```
      syncKeyStores(appStripe='system', keystoreFormat='KSS')
      ```

      For example:

      ```
      connect('weblogic','password','t3://IGDADMINVHN.example.coml:7101')
      svc = getOpssService(name='KeyStoreService')
      svc.importKeyStoreCertificate(appStripe='system',name='trust',password='
      ', keypassword='',alias='bi.example.com',type='TrustedCertificate',
      filepath='/u01/oracle/config/keystores/bi.example.com.pem')
      syncKeyStores(appStripe='system',keystoreFormat='KSS')
      exit()
      ```

You will need to restart the domain for the changes to take effect. The default password for the JDK is *changeit*. The default password for the Node Manager keystores is *COMMON_IAM_PASSWORD*. You will be prompted to confirm that the certificate is valid.

18

Configuring Multi-Data Center

Multi-Data Centers (MDC) help you distribute load as well as address disaster recovery. This chapter provides detailed instructions to help you configure multi- data centers for your enterprise deployment.

Topics

- Variables Used When Configuring Multi-Data Center
  This topic lists the variables used while configuring the Multi-Data Center.
- Roadmap for Configuring Multi-Data Center Deployment
  The roadmap in this section includes the high-level steps for configuring a multi-data center deployment.
- Procuring Resources for a Multi-Data Center Deployment
  You must procure the resources for a multi-data center enterprise deployment.
- Preparing the Load Balancer for a Multi-Data Center Deployment
  A local load balancer configuration for a Multi-Data Center Enterprise Deployment needs to be configured for Site–1 and Site–2.
- Preparing the File System for a Multi-Data Center Deployment
  You must configure the file system for a multi-data center enterprise deployment.
- Preparing the Host Computers for a Multi-Data Center Enterprise Deployment
- Preparing the Database for a Multi-Data Center Deployment
  You must configure the database for a multi-data center enterprise deployment.
- Configuring Oracle LDAP for a Multi-Data Center Deployment
  You must configure Oracle LDAP for a Multi-Data Center Enterprise Deployment.
- Configuring the Web Tier for a Multi-Data Center Deployment
  You must configure the web tier for a multi-data center enterprise deployment.
- Creating the Oracle Access Management Infrastructure for a Multi-Data Center Deployment
- Configuring Oracle Access Management for a Multi-Data Center Deployment
  You must configure Oracle Access Management for a multi-data center enterprise deployment.
- Creating the Oracle Identity Governance Infrastructure for a Multi-Data Center Deployment
- Configuring Oracle Identity Governance for a Multi-Data Center Deployment
  You must configure Oracle Identity Governance for a multi-data center enterprise deployment.
- Updating TAP Endpoint
  The procedure for updating TAP endpoint is explained in this section.
- Enabling Multi-Data Center
  You must enable the Multi-Data Center after building a Multi-Data Center Deployment.

Variables Used When Configuring Multi-Data Center

This topic lists the variables used while configuring the Multi-Data Center.

- *IAD_ORACLE_HOME*
- *IGD_ASERVER_HOME*
- *IAD_ASERVER_HOME*

Roadmap for Configuring Multi-Data Center Deployment

The roadmap in this section includes the high-level steps for configuring a multi-data center deployment.

The steps described in this chapter, to configure multi-data center enterprise deployment, assumes that you have copies of the binaries on both site-1 and site-2. The binaries should be at the same patch level version. Some of the methods of doing this are:

- Manually installing the products on site–1 and site–2
- Remote mirror from site-1 to site-2
- T2P copy and paste method

Creating the binaries is outside of the scope of this chapter. Manual installation steps are described in the earlier chapters of this guide. Remote mirroring is achieved using the mechanisms available within your storage hardware. T2P can be used by following the Oracle T2P documentation.

Table 18-1 Roadmap for Configuring a Multi-Data Center Enterprise Deployment

Task	Sub-Tasks	Description
Configure Site-1 by completing the sub-tasks on Site-1. For each of the sub-tasks, follow the instructions described under Site-1 title, in the respective sections provided in the **Description** column.	Procure the necessary resources.	See Procuring Resources for a Multi-Data Center Deployment.
NA	Preparing the load balancer.	See Preparing the Load Balancer for a Multi-Data Center Deployment.
NA	Prepare the file system.	See Preparing the File System for a Multi-Data Center Deployment.
NA	Prepare the host computers.	See Preparing the Host Computers for a Multi-Data Center Enterprise Deployment.
NA	Prepare the database.	See Preparing the Host Computers for a Multi-Data Center Enterprise Deployment.
NA	Configure Oracle LDAP.	See Configuring Oracle LDAP for a Multi-Data Center Deployment.

Table 18-1 (Cont.) Roadmap for Configuring a Multi-Data Center Enterprise Deployment

Task	Sub-Tasks	Description
NA	Configure Oracle HTTP Server.	See Configuring the Web Tier for a Multi-Data Center Deployment.
NA	Create the Infrastructure for Oracle Access Management.	See Creating the Oracle Access Management Infrastructure for a Multi-Data Center Deployment.
NA	Configure Oracle Access Management.	See Configuring Oracle Access Management for a Multi-Data Center Deployment.
NA	Create the Infrastructure for Oracle Identity Governance.	See Creating the Oracle Identity Governance Infrastructure for a Multi-Data Center Deployment.
NA	Configure Oracle Identity Governance.	See Configuring Oracle Identity Governance for a Multi-Data Center Deployment.
Configure Site-2 by completing the sub-tasks on Site-2. For each of the sub-tasks, follow the instructions described under Site-2 title, in the respective sections provided in the **Description** column.	Procure the necessary resources.	See Procuring Resources for a Multi-Data Center Deployment.
NA	Preparing the load balancer.	See Preparing the Load Balancer for a Multi-Data Center Deployment.
NA	Prepare the file system.	See Preparing the File System for a Multi-Data Center Deployment.
NA	Prepare the host computers.	See Preparing the Host Computers for a Multi-Data Center Enterprise Deployment.
NA	Prepare the database.	See Preparing the Host Computers for a Multi-Data Center Enterprise Deployment.
NA	Extend the existing Oracle LDAP directory to Site-2.	See Configuring Oracle LDAP for a Multi-Data Center Deployment.
NA	Configure Oracle HTTP Server.	See Configuring the Web Tier for a Multi-Data Center Deployment.
NA	Create the Infrastructure for Oracle Access Management.	See Creating the Oracle Access Management Infrastructure for a Multi-Data Center Deployment.
NA	Configure Oracle Access Management.	See Configuring Oracle Access Management for a Multi-Data Center Deployment.
NA	Create the Infrastructure for Oracle Identity Governance.	See Creating the Oracle Identity Governance Infrastructure for a Multi-Data Center Deployment.
NA	Extend the Oracle Identity Governance installation to encompass Site-2.	See Configuring Oracle Identity Governance for a Multi-Data Center Deployment.

Table 18-1 (Cont.) Roadmap for Configuring a Multi-Data Center Enterprise Deployment

Task	Sub-Tasks	Description
Enable Oracle Policy Replication between Site-1 and Site-2.	NA	See Enabling Multi-Data Center.

Procuring Resources for a Multi-Data Center Deployment

You must procure the resources for a multi-data center enterprise deployment.

The procedure for procuring the resources for a multi-data center enterprise deployment is same as the one described in Procuring Resources for an Enterprise Deployment, but with the following additional considerations.

Site–1

When allocating virtual IP addresses, you must ensure that:

- You will need a virtual IP address for the Oracle Access Management Domain on Site 1.
- The virtual IP address you use for the Oracle Identity Manager Domain must be movable to Site 2, if Site–1 becomes unavailable. This allows the Administration server to be started on Site–2, if necessary.

Site–2

When allocating virtual IP addresses, you must ensure that:

- You will need a virtual IP address for the Oracle Access Management Domain on Site 2.
- The virtual IP Address you use for the Oracle Identity Manager Domain must be movable from Site–1, if Site–2 becomes available. This allows the Administration server to be started on Site–1, if necessary.

Preparing the Load Balancer for a Multi-Data Center Deployment

A local load balancer configuration for a Multi-Data Center Enterprise Deployment needs to be configured for Site–1 and Site–2.

The procedure for preparing the load balancer for a multi-data center enterprise deployment is same as the one described in Preparing the Load Balancer and Firewalls for an Enterprise Deployment, but with the following additional considerations.

Site–1

The differential parameters for local load balancer configuration of Site–1 in a Multi-Data Center Enterprise Deployment are:

- The load balancer virtual servers configured within the site will not use the main entry points, that is, `idstore.example.com`, `igdinternal.example.com`, `prov.example.com`, and `login.example.com`, instead, they will use local variants. The names of these entries are not used outside of the load balancer configuration, for example, they can be `login1.example.com` and `prov2.example.com`

- `Iadamin.example.com` will be unique to the deployment, for example: `iadadmin1.example.com`.
- A new load balancer entry point called `oam1.example.com` will be defined, which will contain the Oracle Access Management (OAM) Managed servers in Site–1.
- A global Load balancer virtual host, called `oam.example.com` will be created, which will pass on requests to the local load balancer virtual host `oam1.example.com`.
- The main application entry points, that is, `idstore.example.com` , `login.example.com`, `prov.example.com` and `igdinternal.example.com` will be configured as part of a global load balancer. These entry points will in turn pass on requests to the local load balancer virtual host. For example `prov.example.com` will direct requests to `prov1.example.com`.

Site–2

The differential parameters for local load balancer configuration of Site–2 in a Multi-Data Center Enterprise Deployment are:

- The load balancer virtual servers configured within the site will not use the main entry points, that is, `prov.example.com` and `login.example.com`, instead, they will use local variants. The names of these entries are not used outside of the load balancer configuration, for example, they can be `login2.example.com` and `prov2.example.com`
- `Iadamin.example.com` will be unique to the deployment, for example: `iadadmin2.example.com`.
- A new load balancer entry point called `oam2.example.com` will be defined, which will contain the Oracle Access Management (OAM) Managed servers in Site-2.
- The local load balancer virtual host `oam2.example.com` will be added to the global load balancer definition for `oam.example.com`. Geographic rules will be set up to ensure that invocations originating in Site–1 are sent to `oam1.example.com` and invocations originating in Site–2 are sent to `oam2.example.com`. If the target for the invocations is unavailable, they can be directed to any site which is available..
- The local load balancer virtual hosts for i`idstore2.example.com`, `igdinternal2.example.com`, `prov2.example.com`, and `login2.example.com` will be added to the corresponding global load balancer definition, and geographic rules set up to ensure:
  - Internal traffic originating from a given site is sent to back to that site, if it is available. If the site is not available, internal traffic is directed to another site.
  - Public traffic is directed to the local load balancer that is nearest to their geographical location.

Preparing the File System for a Multi-Data Center Deployment

You must configure the file system for a multi-data center enterprise deployment.

The procedure for preparing the file system for a multi-data center enterprise deployment is same as the one described in Preparing the File System for an Enterprise Deployment, but with the following additional considerations.

Site–1

The differential parameters for file system configuration of Site–1 in a Multi-Data Center Enterprise Deployment are:

- The binaries can optionally be mirrored to Site–2 to prevent the need to keep both sites at the exact same software versions. If doing so, it is important that at least two versions of the binaries are used, as described in the main chapter.
- *IAD_ASERVER_HOME* will be configured locally to Site–1.
- *IAD_ASERVER_HOME* will be mirrored to Site–2 to allow the Governance Administration server to be started on Site–2, if necessary.
- If you are using the file system for whole server migration, then this will need mirroring to Site 2.

Site–2

The differential parameters for file system configuration of Site–2 in a Multi-Data Center Enterprise Deployment are:

- The binaries can optionally be mirrored to Site 1 to prevent the need to keep both sites at the exact same software versions. If doing so, it is important that at least two versions of the binaries are used, as described in the main chapter.
- *IGD_ASERVER_HOME* will be configured locally to Site–2.
- *IGD_ASERVER_HOME* will be mirrored to Site 1 to allow the Governance Admin server to be started on Site–2, if necessary.
- If you are using the file system for whole server migration, then this will need mirroring to Site–1.

Preparing the Host Computers for a Multi-Data Center Enterprise Deployment

The procedure for preparing the host computers for a multi-data center enterprise deployment is same as the one described in Preparing the Host Computers for an Enterprise Deployment.

Preparing the Database for a Multi-Data Center Deployment

You must configure the database for a multi-data center enterprise deployment.

The procedure for preparing the database for a multi-data center enterprise deployment is same as the one described in Preparing the Database for an Enterprise Deployment, but with the following additional considerations.

Site–1

The observations to be made when setting up a database for a multi-data center enterprise deployment for Site–1 are:

- A dedicated database will be used for Oracle Access Management (OAM) for Site–1.
- A dedicated database will be used for Oracle Identity Governance (OIG) for Site–1.
- Role-based database services must be used for OIG.

Site–2

The observations to be made when setting up a database for a multi-data center enterprise deployment for Site–2 are:

- A dedicated database will be used for Oracle Access Management (OAM) for Site–2.

- Data guard will be used to replicate the Oracle Identity Governance (OIG) database to Site–2
- Role-based database services must be used for OIG and configured on both Site–1 and Site–2.

Configuring Oracle LDAP for a Multi-Data Center Deployment

You must configure Oracle LDAP for a Multi-Data Center Enterprise Deployment.

The Oracle LDAP configuration for a Multi-Data Center Enterprise Deployment for Site–1 and Site–2 is described in the following sections.

Site–1

Configure LDAP on Site–1 as described in Configuring Oracle LDAP for an Enterprise Deployment.

Site–2

LDAP is configured on Site–2 as an extension of Site–1. Therefore, you should configure LDAP on Site–2 as follows:

- For Oracle Unified Directory (OUD), complete the following tasks:
  – Configuring Oracle Unified Directory Instance on LDAPHOST2
  – Validating Oracle Unified Directory on LDAPHOST2
  – Granting OUD changelog Access
  – Updating Oracle Unified Directory ACIs
  – Creating OUD Indexes

Configuring the Web Tier for a Multi-Data Center Deployment

You must configure the web tier for a multi-data center enterprise deployment.

Configure Oracle HTTP Server (as described in Configuring Oracle HTTP Server for an Enterprise Deployment), but with the following additional considerations:

Site-1

The observations to be made when configuring the web tier for a Multi-Data Center Enterprise Deployment for Site 1 are:

- When creating the virtual hosts (either Oracle HTTP Server or Oracle Internet Directory), ensure that the main entry points are used, that is, `prov.example.com` , `login.example.com`, and `igdinternal.example.com`rather than the local variations.
- `Iadadmin1.example.com` should still be used for the `iadadmin.example.com`entry point. `Iadadmin1.example.com` will always be used to manage the Oracle Access Management (OAM) domain on Site–1.
- Optionally, include the Weblogic cluster members from Site–2 in the virtual host definitions. If the managed servers in Site 1 are down, but the web tier in Site–1 is functioning, then Site 1 can direct requests to managed servers on Site–2. While this is a legitimate use case, it is more often not desirable to direct requests to managed servers to prevent traffic from bouncing between the sites. If you do not wish traffic bouncing between the sites, then you need to add the Oracle HTTP Server directive DynamicServerList OFF.

```
# OIM Self Service
<Location/Identity>

SetHandler webLogic-handler

WebLogicCluster PMHOST1.example.com:14000, OIMHOST2.example.com:14000

DynamicServerList OFF

</Location
```

Site–2

The observations to be made when configuring the web tier for a Multi-Data Center Enterprise Deployment for Site–2 are:

- When creating the virtual hosts (either Oracle HTTP Server or Oracle Internet Directory), ensure that the main entry points are used, that is, `prov.example.com` , `login.example.com`, and `igdinternal.example.com` rather than the local variations.
- `Iadadmin2.example.com` should still be used for the `iadadmin.example.com` entry point. `Iadadmin2.example.com` will always be used to manage the Oracle Access Management (OAM) domain on Site–2.
- Optionally, include the Weblogic cluster members from Site–2 in the virtual host definitions. If the managed servers in Site–1 are down, but the Web Tier in Site–1 is functioning, then Site–1 can direct requests to managed servers on Site–2. While this is a legitimate use case, it is more often not desirable to direct requests to managed servers to prevent traffic from bouncing between the sites. If you do not wish traffic bouncing between the sites, then you need to add the Oracle HTTP Server directive DynamicServerList OFF.

```
# OIM Self Service
<Location/Identity>

SetHandler webLogic-handler

WebLogicCluster OIMHOST3.example.com:14000, OIMHOST2.example.com:14000

DynamicServerList OFF

</Location
```

Be aware that if you are using dynamic clusters with OIM, the port numbers will be an increment of those on Site–1. For example: Site 1 (14000, 14001) and Site–2 (14002, 14003).

Creating the Oracle Access Management Infrastructure for a Multi-Data Center Deployment

The creation of the Oracle Access Management Infrastructure for a multi-data center enterprise deployment for Site–1 and Site –2 is the same as the standard EDG process described in Creating Infrastructure for Oracle Access Management.

Configuring Oracle Access Management for a Multi-Data Center Deployment

You must configure Oracle Access Management for a multi-data center enterprise deployment.

The procedure for configuring Oracle Access Management for a multi-data center enterprise deployment is same as the one described in Configuring Oracle Access Management, but with the following additional considerations.

Site–1

After configuring OAM, complete the following tasks:

1. Create a Server Instance for Load Balancer

   When OAM is initially configured, a number of OAM server instances would have been created for each managed server. These server instances are used to determine where to send OAM NAP calls. In a Multi-Data Center deployment, you need to create an additional server instance for the load balancer entry point, `oam.example.com`. For more information on how to create a server instance, refer Creating an Additional Server Instance for the Load Balancer.

2. Update Webgate Agents to Use load Balancer

   Update the Webgate Agents to use the load balancer entry `oam.example.com` rather than the named server names. For more information on how to To update the Webgate Agents, refer Updating the Webgate Agents to Use Load Balancer

Site–2

The configuration of OAM for Site–2 is similar to that described in the configuration for Site–1. The updating of webgate agents to use the load balancer entry point is taken care of with policy synchronisation.

- Creating an Additional Server Instance for the Load Balancer
  You must create an additional server instance for the load balancer entry point in a Multi-Data Center Enterprise Deployment.
- Updating the Webgate Agents to Use Load Balancer
  You must update the Webgate Agents for the load balancer entry point in a Multi-Data Center Enterprise Deployment.

Creating an Additional Server Instance for the Load Balancer

You must create an additional server instance for the load balancer entry point in a Multi-Data Center Enterprise Deployment.

To create an additional server instance, perform the following steps:

1. Log in to the OAM console using the URL as oamadmin user that you have created in the EDG.

   `http://iadadmin.example.com/oamconsole`

2. Click the **Configuration** tab.
3. Click **Server Instances**.
4. On the Search OAM Servers page, click **Search**.

   The server instances that are already defined is displayed.

5. Click **Create**
6. Enter the server name as `oamLBR`
7. Enter the loadbalancer virtual host name. For example: `oam.example.com`

8. Enter the port that you have configured on the loadbalancer virtual host to listen on. For example: 5575
9. Enter the proxy server ID that you have used in the existing server entries. For example: `AccessServerConfigProxy`
10. Select the OAM security model that you are using. For example: Simple
11. Click **Apply**.

Updating the Webgate Agents to Use Load Balancer

You must update the Webgate Agents for the load balancer entry point in a Multi-Data Center Enterprise Deployment.

To update the Webgate Agents, do the following:

1. Click an agent name.
2. In the **Primary Server** box, click the **Add** button to create a new agent, and select `oamLBR` as the access server.
3. Delete each of the other entries in the Primary Server List.
4. Click **Apply**
5. Repeat these steps for each of the agents.

Creating the Oracle Identity Governance Infrastructure for a Multi-Data Center Deployment

The procedure for configuring Oracle Identity Governance (OIG) Infrastructure for a multi-data center enterprise deployment for Site–1 and Site— 2 is largely the same as the standard EDG process described in Creating Infrastructure for Oracle Identity Governance. There are however a few exceptions which are described below.

Configuring Oracle Identity Governance for a Multi-Data Center Deployment

You must configure Oracle Identity Governance for a multi-data center enterprise deployment.

The configuration of Oracle Identity Governance (OIG) for a multi-data center enterprise deployment is the same as the standard EDG process described in Configuring Oracle Identity Governance. In addition to this, perform the additional tasks for Site-1 and Site-2.

Site–1

After you configure Oracle Identity Governance on Site-1, perform the following tasks on Site-1:

1. Configuration Wizard changes for Oracle Identity Manager
2. Post-configuration Changes for Oracle Identity Manager

Site–2

After you configure Oracle Identity Governance on Site-2, disable the OIM job scheduler for Site–2 as described in Disabling the Oracle Identity Manager Job Scheduler for Configuring the Oracle Identity Manager.

- Configuration Wizard changes for Oracle Identity Manager
- Post-configuration Changes for Oracle Identity Manager
- Disabling the Oracle Identity Manager Job Scheduler for Configuring the Oracle Identity Manager

Configuration Wizard changes for Oracle Identity Manager

The following additions should be made in the configuration wizard when configuring Oracle Identity Manager this will create a cluster which spans the two sites:

- Oracle Identity Manager (OIM) Cluster Configuration

  In the configuration wizard, while creating the IAMGovernanceDomain, you must include ALL servers (Site 1 and Site 2), when specifying the OIM and SOA clusters. For example, `OIMHOST1.example.com`:14000, `OIMHOST2.example.com`:14000, `OIMHOST3.example.com:14000`, `OIMHOST4.example.com`:14000

  For dynamic clusters, you must include only the machines for Site 1 and Site 2.

  If you have an existing deployment, you can use the standard scale up/out procedures to include the extra cluster members.

- JMS

  Configure JMS to use the database by creating JMS stores for All the OIM/SOA managed servers.

- Data sources

  When creating the JMS data sources, you must specify the OIM service name and not the database service name; this will be a role-based database service.

Post-configuration Changes for Oracle Identity Manager

The following post-configuration changes should be made when configuring Oracle Identity Manager (OIM):

- Update OIM data source

  When the OIM domain was created, a number of datasources would have been created that use the `oim.example.com` database service. All these services will be pointing at the primary OIM database. In order to facilitate a seamless failover, the standby database needs to be added in to the data source configurations. A typical database connection which includes both a primary and standby database looks like:

  ```
  jdbc:oracle:thin:@ (DESCRIPTION_LIST=(LOAD_BALANCE=OFF)(FAILOVER=ON)
  (DESCRIPTION=(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
  (ADDRESS_LIST=(LOAD_BALANCE=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=iagdb-
  scan.example.com)(PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=oim.example.com)))
  (DESCRIPTION=(CONNECT_TIMEOUT=3)(RETRY_COUNT=3)
  (ADDRESS_LIST=(LOAD_BALANCE=ON)(ADDRESS=(PROTOCOL=TCP)(HOST=iagdbdg-
  ```

```
scan.example.com)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=oim.example.com))))
```

These data sources needs to be changed. For more information on how to change the data sources, refer Changing the Oracle Identity Manager Data Sources for Configuring Oracle Identity Manager

- Update Java Virtual Machine Process Status Tool (JPS) security files

  In addition to updating the weblogic data sources, you must also update the data source information in the files, jps-config.xml and jps-config-jse.xml, which are located in the directory *IGD_ASERVER_HOME/config/fmwconfig*.

  > **Note:**
  >
  > You must take a back up of these files before editing them.

- Changing the Oracle Identity Manager Data Sources for Configuring Oracle Identity Manager

Changing the Oracle Identity Manager Data Sources for Configuring Oracle Identity Manager

You must change the Oracle Identity Manager data sources when performing the post-configuration changes for configuring Oracle Identity Manager (OIM). To change the OIM data sources, do the following:

1. Login to the WebLogic Console.
2. Click **Lock and Edit**.
3. In the Domain Tree, expand **Services** – **Data Sources**.
4. On the Summary of JDBC Data sources page, click the name of a data source. For example, click ApplicationDB.
5. Click the **Connection Pool** tab.
6. Update the URL to reflect the above database connection example.
7. Click **Save**.
8. Validate the data source by clicking the **Monitoring** tab and the **Testing** sub tab.
9. Select one of the servers and click **Test Data Source**.

   You make sure the test is successful before continuing.
10. Click the **ONS** tab.
11. Add the standby database to the ONS Nodes field by separating each host/port with a comma. For example: `primaryDB-scan:6200,standbyDB-scan:6200`.
12. Click **Save**.
13. Repeat for each data source.
14. Click **Activate Changes**.

Disabling the Oracle Identity Manager Job Scheduler for Configuring the Oracle Identity Manager

You must the disable the Oracle Identity Manager job scheduler on Site 2 for configuring the Oracle Identity Manager.

The OIM job scheduler uses the database intensively. In order to keep inter-site traffic to a minimum, the job scheduler should be disabled on the site, where the database is not primary. This step is not necessary, if you plan to shutdown the OIM Managed servers on Site 2.

The jobscheduler is disabled by adding the following parameter to the server startup arguments:

```
-Dscheduler.disabled=true
```

For more information on how to disable the job scheduler, refer Adding the Parameter to Disable Job Scheduler for Configuring Oracle Identity Manager

- Adding the Parameter to Disable Job Scheduler for Configuring Oracle Identity Manager

Adding the Parameter to Disable Job Scheduler for Configuring Oracle Identity Manager

To add the parameter, `Disabling th`, to disable the job scheduler, do the following:

1. Log in to the WebLogic Administrative Console.
2. In left pane, click **Environment**, **Servers**.
3. Click **Lock and Edit** in the left tab.
4. Click **Configuration**, **Server start** tab in the right pane.
5. In the **Argument** text box, add `–Dscheduler.disabled=true,` and save.
6. Click **Activate Change** in the left pane.

   After switching over the database, this parameter should be removed from these servers and added into the server definitions of the now standby site.

Updating TAP Endpoint

The procedure for updating TAP endpoint is explained in this section.

If you have configured the OAM MDC prior to starting the configuration of Oracle Identity Governance then this step is not required. If you are converting an existing OAM to a multi-datacenter, then you may need to update the TAP endpoint in OIM to reflect the new OAM Entry point:

1. Log in to the **IdentityAccessDomain** in the Oracle Enterprise Manager using the user `weblogic_idm`.

   ```
   http://iadadmin.example.com/em
   ```

2. Click **weblogic_domain > System Mbean Browser**.
3. In the search box, enter `oamWLST`, and click **Search**. The mbean is displayed.

4. Select the `Operations` tab.
5. Click `displayTAPURL`, make a note of the return value. Check the example mentioned below.

   ```
   https://login.example.com:443/oam/server/dap/cred_submit
   ```

6. Log in to the **IdentityGovernanceDomain** of Oracle Enterprise Manager using the user `weblogic_idm`.

   ```
   http://igdadmin.example.com/em
   ```

7. Click **weblogic_domain > System Mbean Browser**.
8. In the search box, enter `SSOIntegrationMXBean`, and click **Search**. The mbean is displayed.
9. Set the value of `TapEndpointUrl` attribute to the value noted from `oamWLST` bean.
10. Click **Apply**.

Enabling Multi-Data Center

You must enable the Multi-Data Center after building a Multi-Data Center Deployment.

You can enable the Multi-Data Center by performing the following steps. These steps are explained in the following sections.

- Setting up a Primary Data Center
- Setting up a Clone Data Center
- Enabling Replication

Setting up a Primary Data Center

To set up a Primary Data Center, you must configure a Primary Data Center using MDC ADMIN REST APIs.

Run the following command with appropriate values to configure the Primary Data Center.

```
curl -k -u oamadmin:password -H 'Content-Type: application/json' -X POST
'http://MasterServerURL/oam/services/rest/mdc/master' -d
'{"mdcTopologyType":"value",
"masterMDCAgentID":"value","cloneMDCAgentID":"value",
"accessClientPassword":"value","artifactPassword":"value","cloneServerURL":"va
lue","agentKeyPassword":"value","certModeKeystorePassword":"value","masterServ
erURL":"value",
"cloneAdminUserNamePassword":"value","trustStorePath":"value",
"keyStorePath":"value", "artifactsZipLocation":"value"}'
```

The following table describes the values for the Curl command.

Table 18-2   Configuring the Primary Data Center

Value	Description
-u	The user name and password of the OAM Administrative user. For example: iamadmin

Table 18-2 (Cont.) Configuring the Primary Data Center

Value	Description
MasterServerURL	The URL of the OAM domain you are designating as primary. For example: http://iadadmin1.example.com/oam/services/rest/mdc/master
mdcTopologyType	The two topology types available for MDC configuration, ACTIVE_ACTIVE or DISASTER_RECOVERY. In this case, choose ACTIVE_ACTIVE.
masterMDCAgentID	The MDC NAP Agent Name for the Primary Data Center. For, enter Site1
cloneMDCAgentID	The MDC NAP Agent Name for the Clone data center. For example, enter Site2
accessClientPassword	The password required to be used by the MDC NAP agents in Primary and Clone data centers
artifactPassword	The password that is used to protect cloning artifacts
cloneServerURL	The URL of the Primary Admin server or the URL of the reverse proxy front ending the Primary Admin Server. For example, enter http://iamadmin1.example.com/
masterServerURL	The URL of the Primary Admin server or the URL of the reverse proxy front ending the Primary Admin Server. For example, enter http://iamadmin1.example.com/
cloneAdminUserNamePassword	The user credentials of the Clone data center's IAM Administrator if the user name and password of the Administrator for Primary and Clone data centers are different. For example, enter `iamadmin`
trustStorePath	The path to oamclient-truststore.jks file. For example *IAD_ASERVER_HOME*/output/webgate-ssl-SHA-256/
keyStorePath	The path to oamclient-keystore.jks file file For example *IAD_ASERVER_HOME*/output/webgate-ssl-SHA-256/
artifactsZipLocation	The location where cloning artifacts are to be stored; specify only if cloning artifacts need to be stored in any location other than /tmp. For example *SHARED_CONFIG_DIR*/mdc

> **Note:**
>
> Before running the Curl command, make sure that the location specified in artifactsZipLocation already exists.

A sample Curl command for configuring a Primary Data Center is as follows:

```
curl -k -u oamadmin:password -H 'Content-Type: application/json' -X POST
'http://iadadmin1.example.com/oam/services/rest/mdc/master' -d
'{"mdcTopologyType":"ACTIVE_ACTIVE",
"masterMDCAgentID":"site1Agent","cloneMDCAgentID":"site2Agent",
"accessClientPassword":"password","artifactPassword":"password","cloneServerUR
L":"http://
iadadmin2.example.com","agentKeyPassword":"password","certModeKeystorePassword
":"password","masterServerURL":"http://iadadmin1.example.com",
"cloneAdminUserNamePassword":"password","trustStorePath":"/u01/oracle/config/
domains/IAMAccessDomain/output/webgate-ssl-SHA-256", "keyStorePath":"":"/u01/
oracle/config/domains/IAMAccessDomain/output/webgate-ssl-SHA-256",
"artifactsZipLocation":"/u01/oracle/config/mdc"}'
```

After this command is executed, two new agents are created in the OAM Console called, site1Agent and site2Agent. Verify that these agents reference the load balancer entry for OAM by following the steps in Updating the Webgate Agents to Use Load Balancer section.

Setting up a Clone Data Center

To set up a Clone Data Center, you must configure a Clone Data Center for Multi-Data Center (MDC) environment using MDC ADMIN REST APIs as follows:

Run the following command with appropriate values to configure the Clone Data Center.

```
curl -k -u oamadmin:password -H 'Content-Type: application/json' -X POST
'http://CloneServerURL/oam/services/rest/mdc/clone' -d
'{"masterServerURL":"value","artifactPassword":"value","masterAdminUserNamePas
sword":"value", "artifactsZipLocation":"value",
"masterArtifactsZipLocation":"value"}'
```

The following table describes the values for the Curl command.

Table 18-3 Clone Data Center Properties

Value	Description
-u	The user name and password of the OAM Administrative user. For example: iamadmin
CloneServerURL	The URL of the OAM domain you are designating as a clone. For example: http://iadadmin2.example.com/oam/services/rest/mdc/clone
masterServerURL	The URL of the Primary Admin server or the URL of the reverse proxy front ending the Primary Admin Server. For example, enter http://iamadmin.example.com/
artifactPassword	The same password that protects cloning artifacts and is used while setting up the Primary data centers
masterAdminUserNamePassword	The user credentials of the Primary data center’s IAM Administrator
artifactsZipLocation	The location where cloning artifacts are to be stored; specify only if cloning artifacts need to be stored in any location other than /tmp. For example *SHARED_CONFIG_DIR*/mdc

Note:

Before running the Curl command, make sure that the location specified in artifactsZipLocation already exists on the clone admin server machine.

A sample Curl command for configuring a Clone Data Center is as follows:

```
curl -k -u oamadmin:password -H 'Content-Type: application/json' -X POST
'http://iadadmin2.example.com/oam/services/rest/mdc/clone/configuration'
```

In a Multi-Data Center deployment, only one site is responsible for the creation of policies; the site that been designated as primary site, as explained in the section Setting up a Primary Data Center. To prevent inadvertent writes to the Clone site, you must place the Clone site into read-only mode.

To place the clone site into read-only mode on the clone site, do the following:

```
IAD_ORACLE_HOME/oracle_common/common/bin/wlst.sh
connect('weblogic','password','t3://iadadmminvhn2.example.com:7001')
domainRuntime()
setMultiDataCenterWrite(WriteEnabledFlag="true")
exit()
```

After setting up the Clone Data Center, you must restart the domain and validate the configuration, for Site 1 and Site 2.

For more information, refer the following topics:

- Restarting the Domains
- Validating the Configuration of Clone Data Center
- Restarting the Domains
- Validating the Configuration of Clone Data Center

Restarting the Domains

After setting up the Clone Data Center, you must restart the domains for Site 1 and Site 2.

To restart the domains on Site 1 and Site 2, do the following:

1. Shutdown the Admin Server and Managed Servers on Site 1.
2. Shutdown the Admin Server and Managed Servers on Site 2.
3. Startup the Admin Server and Managed Servers on Site 1.
4. Startup the Admin Server and Managed Servers on Site 2.

Validating the Configuration of Clone Data Center

After setting up the Clone Data Center, you must validate the configuration of Clone Data Center for Site 1 and Site 2.

To validate the configuration, execute the following command:

```
curl -k -u oamadmin:XXXX 'http://SiteURL/oam/services/rest/mdc/configuration'
```

For Site 1, the command is:

```
curl -k -u oamadmin:password “http://iadadmin1.example.com/oam/services/
rest/mdc/configuration”
```

For Site 1, the command is:

```
curl -k -u oamadmin:password -H 'Content-Type: application/json' -X POST
“http://iadadmin2.example.com/oam/services/rest/mdc/configuration”
```

Enabling Replication

After the configuration has been placed into Multi-Data Center mode you have to set up data synchronization. This set-up ensures that any information added to the Primary site is propagated to the Clone sites.

The set-up is a two-stage process, which are:

- To perform a bulk data unload from the Primary site to the Clone site
- To create a replication agreement so that subsequent changes are propagated automatically

For more information, refer the following topics:

- Performing a Bulk Data Unload from the Primary Site to the Clone Site
- Creating a Replication Agreement

Performing a Bulk Data Unload from the Primary Site to the Clone Site

To perform a bulk data unload from the Primary site to the Clone site, you must refresh the Clone with the Primary site's Data, which is achieved by exporting the Access Store from the Primary and importing it into the clone.

To refresh the Clone with the Primary site's Data, issue the following command on the Primary Node:

```
IAD_ORACLE_HOME/oracle_common/common/bin/wlst.sh
connect('weblogic','password','t3://iadadmin1vhn.example.com:7001')
exportAccessStore(toFile=”filelocation/oamaccess.zip",namePath="/")
exit()
```

> **Note:**
>
> A file location must exist for the command to be executed.

Copy the generated file to the Clone admin server host.

To refresh the Clone with the Primary site's Data, issue the following command on the Clone Node:

```
IAD_ORACLE_HOME/oracle_common/common/bin/wlst.sh
connect('weblogic','password','t3://iadadmin1vhn.example.com:7001')
importAccessStore(toFile=”filelocation/oamaccess.zip",namePath="/")
exit()
```

Creating a Replication Agreement

The final step in the process is to set up a replication agreement between the Primary and the Clone Sites. But, before creating a replication agreement, it is good practice to perform the processes listed below. These processes are explained in the following sections.

- Validate the Replication End Points

- Obtain the MDC Cluster Names
- Encode the iamadmin User

To set up a replication agreement between the Primary and the Clone Sites, execute the following command:

```
curl -k -u oamadmin:password -H 'Content-Type: application/json' -X POST
'http://masterSiteURL/oam/services/rest/_replication/setup' -d
'{"name":”agreementName”,"source":”masterClusterName”,"target":”cloneClusterNa
me”,"documentType":"ENTITY","config": {"entry":
{"key":"authorization","value":”Basic encodedUser” }}}'
```

The following table describes the values for the Curl command.

Table 18-4   Values for the Curl Command

Value	Description
-u	The user name and password of the OAM Administrative user. For example: iamadmin
masterSiteURL	The URL of the OAM domain you are designating as primary, for example: http://iadadmin1.example.com/ oam/services/rest/ _replication/setup
agreementName	A name of the agreement of your choice. For example: Site1toSite2
masterClusterName	The name of the primary sites cluster obtained from Obtain MDC Cluster Names
clonerClusterName	The name of the cluster sites cluster obtained from Obtain MDC Cluster Names
encodedUser	The iamadmin encoded user string as obtained in Encode the iammadmin User

A sample Curl command for creating a replication agreement is as follows

```
curl -k -u iamadmin:password -H 'Content-Type: application/json' -X POST
'http://iadadmin1.example.com/oam/services/rest/_replication/setup' -d
'{"name":"Site1toSite2","source":"0c04b-OAMHOST1.u","target":"0c04b-
OAMHOST2.u","documentType":"ENTITY","config": {"entry":
{"key":"authorization","value":"Basic aWFtYWRtaW46UGFzc3dvcmQxr" }}}'
```

If the replication is successful, the output is similar to as shown below:

```
{"enabled":true,"identifier":"201709071449437364","lastSequenceNumber":878,"ok":true,"pollInterval":900,"startingSequenceNumber":878,"state":"READY"}
```

Note:

The POLLINTERVAL is 900 sec (15 min). Hence the changes made in Primary DC takes 15 minutes to get propagated to Clone DC. If the poll interval is too long, you can modify the poll interval in Clone DC.

To modify the poll interval in Clone DC, do the following:

1. Obtain the replication Identifier (replId) by sending a query to the existing replication agreements using the following command:

   ```
   curl -k -u oamadmin:password 'http://iadadmin1.example.com/oam/services/
   rest/_replication/agreements'
   ```

   If there are multiple agreements, you must select the identifier for which APS has to be disabled by executing the same command on the corresponding Clone Data Center.

2. Execute the following command to modify poll interval in Clone DC:

   ```
   curl -u weblogic:password -H 'Content-Type: application/json' -X PUT
   'http://iamadmin1.example.com/oam/services/rest/_replication/
   replicationAgreement' -d
   '{"pollInterval":PollInterval,"replicaType":"CONSUMER"}'
   ```

   Where

   - replicationAgreement is the value obtained in step 1
   - Pollinterval is the new polling interval in seconds

- Validating the Replication End Points
- Obtaining the Multi-Data Center Cluster Names
- Encoding the oamadmin User

Validating the Replication End Points

Before creating a replication agreement, it is good practice to ensure that REST endpoints, which facilitate replication, are working on both the primary and clone sites. If these end points are not working, you will not be able to create a replication agreement.

To validate the replication end points, you must issue the following command:

```
curl -u oamadmin:password 'http://SiteURL/oam/services/rest/_replication/
hello'
```

A sample Curl command for validating replication end points is as follows:

```
curl -u oamadmin:password 'http://iadadmin1.example.com/oam/services/rest/
_replication/hello'


curl -u oamadmin:password 'http:// iadadmin2.example.com /oam/services/rest/
_replication/hello'
```

Obtaining the Multi-Data Center Cluster Names

Each site in the Multi-Data Center deployment has a unique cluster name, which is required when creating a replication agreement.

Query the existing configurations using the following command to determine the cluster name of each site:

```
curl -k -u oamadmin:password 'http://siteURL/oam/services/rest/mdc/dc/
configuration
```

A sample Curl command for Obtaining the Multi-Data Center Cluster Names is as follows:

```
curl -k -u oamadmin:password 'http://iadadmin1.example.com/oam/services/
rest/mdc/dc/configuration
```

```
curl -k -u oamadmin:password 'http://iadadmin2.example.com/oam/services/
rest/mdc/dc/configuration
```

The output is similar to as shown below with the cluster name highlighted:

```
{"ok":true,"status":"Success","clusterName":"0c04b-OAMHOST1.u" ,"primaryServers":
["OAMHOST1.example.com:5575","OAMHOST2.example.com:5575","oamLBR.example.com:5575"],"rest
EndPoint":"http://IADADMINVHN1.example.com:7001","oamServerModes":
{"wls_oam2":"simple","oamLBR":"simple","wls_oam1":"simple"}}
```

Make a note of the cluster names.

Encoding the oamadmin User

When you setup the replication agreement, you need to use the `oamadmin` account, which resides in your LDAP directory.

Setting up the replication agreement is achieved using curl commands. As part of the curl command, you need to supply the oamadmin user and password. This password must be encoded using base64. There are many utilities on the web that allow you to encode the password, one example is http://www.motobit.com/util/base64-decoder-encoder.asp

> **Note:**
>
> When encoding the value you must encode both the username and the password. For example; when using the above utility the value to be encoded should be in the form **username:password** (For example; oamadmin:<*password*>).

Part IV

Common Configuration and Management Procedures for an Enterprise Deployment

There are certain configuration and management procedures that are recommended for a typical enterprise deployment.

The following topics contain configuration and management procedures that are required for a typical enterprise deployment.

- Common Configuration and Management Tasks for an Enterprise Deployment
  The configuration tasks include a few that are common to all enterprise deployments, such as verifying sizing information, performing backups and recoveries, and so on. Patching an enterprise deployment and cross wiring components are the other common tasks.
- Using Whole Server Migration and Service Migration in an Enterprise Deployment
  The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. The following sections explain how these features can be used in an Oracle Fusion Middleware enterprise topology.
- Scaling Procedures for an Enterprise Deployment
  The scaling procedures for an enterprise deployment include scale out, scale in, scale up, and scale down. During a scale-out operation, you add managed servers to new nodes. You can remove these managed servers by performing a scale in operation. During a scale-up operation, you add managed servers to existing hosts. You can remove these servers by performing a scale-down operation.
- Configuring Single Sign-On for an Enterprise Deployment
  You need to configure the Oracle HTTP Server WebGate in order to enable single sign-on with Oracle Access Manager.
- Sanity Checks
- Troubleshooting
  You can troubleshoot the common issues that may arise with the Identity and Access Management enterprise deployment. The solutions provided for the common problems help you resolve them quickly.

19

Common Configuration and Management Tasks for an Enterprise Deployment

The configuration tasks include a few that are common to all enterprise deployments, such as verifying sizing information, performing backups and recoveries, and so on. Patching an enterprise deployment and cross wiring components are the other common tasks.

This chapter includes the following topics:

- Configuration and Management Tasks for All Enterprise Deployments
  Complete these common configuration tasks that apply to any Oracle Fusion Middleware enterprise deployment. These tasks include checking the sizing requirements for the deployment, using the JDBC persistence store for web services, and taking backups of the deployment.
- Configuration and Management Tasks for an Oracle Identity and Access Management Enterprise Deployment
  These are some of the key configuration and management tasks that you likely need to perform on an Oracle Identity and Access Management enterprise deployment.
- Considerations for Cross-Component Wiring
  Cross-Component Wiring (CCW) enables the FMW components to publish and bind to some of the services available in a WLS domain, by using specific APIs.
- Starting and Stopping Servers in Dynamic Clusters
  You can start and stop server instances in dynamic clusters by using the same methods you use to start and stop server instances in configured static clusters.
- 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.

Configuration and Management Tasks for All Enterprise Deployments

Complete these common configuration tasks that apply to any Oracle Fusion Middleware enterprise deployment. These tasks include checking the sizing requirements for the deployment, using the JDBC persistence store for web services, and taking backups of the deployment.

- Verifying Appropriate Sizing and Configuration for the WLSSchemaDataSource
  `WLSSchemaDataSource` is the common datasource that is reserved for use by the FMW components for JMS JDBC Stores, JTA JDBC stores, and Leasing services. `WLSSchemaDataSource` is used to avoid contention in critical WLS infrastructure services and to guard against dead-locks.
- Verifying Manual Failover of the Administration Server
  In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from OIMHOST1 and OIMHOST2 are detailed in the following sections.

- Configuring Listen Addresses in Dynamic Cluster Server Templates
  The default configuration for dynamic managed servers in dynamic clusters is to listen on all available network interfaces. In most cases, this may be undesirable.
- Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment
  After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur. If you implement dynamic clusters, the configuration of the server template assigned to each newly added cluster should be verified and updated, otherwise, verify and update every statically-defined Managed Server for the newly added clusters.
- Setting the Front End Host and Port for a WebLogic Cluster
  You must set the front-end HTTP host and port for the Oracle WebLogic Server cluster that hosts the Oracle Identity and Access Management servers. You can specify these values in the Configuration Wizard while you are specifying the properties of the domain.
- Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer
  It is important to understand how to enable SSL communication between the middle tier and the hardware load balancer.
- Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment
  The persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence.
- About JDBC Persistent Stores for Web Services
  By default, web services use the WebLogic Server default persistent store for persistence. This store provides high-performance storage solution for web services.
- Performing Backups and Recoveries for an Enterprise Deployment
  It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle Identity and Access Management enterprise deployment.

Verifying Appropriate Sizing and Configuration for the WLSSchemaDataSource

`WLSSchemaDataSource` is the common datasource that is reserved for use by the FMW components for JMS JDBC Stores, JTA JDBC stores, and Leasing services. `WLSSchemaDataSource` is used to avoid contention in critical WLS infrastructure services and to guard against dead-locks.

To reduce the `WLSSchemaDataSource` connection usage, you can change the JMS JDBC and TLOG JDBC stores connection caching policy from *Default* to *Minimal* by using the respective connection caching policy settings. When there is a need to reduce connections in the back-end database system, Oracle recommends that you set the caching policy to *Minimal* . Avoid using the caching policy *None* because it causes a potential degradation in performance. For a detailed tuning advice about connections that are used by JDBC stores, see Configuring a JDBC Store Connection Caching Policy in *Administering the WebLogic Persistent Store*.

The default `WLSSchemaDataSource` connection pool size is 75 (size is double in the case of a GridLink DataSource). You can tune this size to a higher value depending on the size of the different FMW clusters and the candidates that are configured for migration. For example, consider a typical SOA EDG deployment with the default number of worker threads per store. If more than 25 JDBC Stores or TLOG-in-DB instances or both can fail over to the same Weblogic server, and the Connection Caching Policy is not changed from *Default* to *Minimal*, possible connection contention issues could arise. In these cases, increasing the default `WLSSchemaDataSource` pool size (maximum capacity) becomes necessary (each JMS store uses a minimum of two connections, and leasing and JTA are also added to compete for the pool).

Verifying Manual Failover of the Administration Server

In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from OIMHOST1 and OIMHOST2 are detailed in the following sections.

Assumptions:

- The Administration Server is configured to listen on ADMINVHN, and not on localhost or on any other host's address.

  For more information about the ADMINVHN virtual IP address, see Reserving the Required IP Addresses for an Enterprise Deployment.
- These procedures assume that the Administration Server domain home (*ASERVER_HOME*) has been mounted on both host computers. This ensures that the Administration Server domain configuration files and the persistent stores are saved on the shared storage device.
- The Administration Server is failed over from OIMHOST1 to OIMHOST2, and the two nodes have these IPs:
  - OIMHOST1: 100.200.140.165
  - OIMHOST2: 100.200.140.205
  - ADMINVHN : 100.200.140.206. This is the Virtual IP where the Administration Server is running, assigned to a virtual sub-interface (for example, eth0:1), to be available on OIMHOST1 or OIMHOST2.
- Oracle WebLogic Server and Oracle Fusion Middleware components have been installed in OAMHOST2 as described in the specific configuration chapters in this guide.

  Specifically, both host computers use the exact same path to reference the binary files in the Oracle home.

The following topics provide details on how to perform a test of the Administration Server failover procedure.

- Validating Access to the Administration Server on OIMHOST2 Through Oracle HTTP Server
  If you have configured the web tier to access AdminServer, it is important to verify that you can access the Administration Server after you perform a manual failover of the Administration Server, by using the standard administration URLs.

Validating Access to the Administration Server on OIMHOST2 Through Oracle HTTP Server

If you have configured the web tier to access AdminServer, it is important to verify that you can access the Administration Server after you perform a manual failover of the Administration Server, by using the standard administration URLs.

From the load balancer, access the following URLs to ensure that you can access the Administration Server when it is running on OIMHOST2:

- `http://admin.example.com/console`

This URL should display the WebLogic Server Administration console.

- `http://admin.example.com/em`

  This URL should display Oracle Enterprise Manager Fusion Middleware Control.

Configuring Listen Addresses in Dynamic Cluster Server Templates

The default configuration for dynamic managed servers in dynamic clusters is to listen on all available network interfaces. In most cases, this may be undesirable.

In preparation for disaster recovery, Oracle recommends that you use host name aliases that can be mapped to different IPs in different data centers (for example, OIMHOST1, OIMHOST2) to set each server's listen address to a specific network interface. With dynamic clusters, each server cannot be configured specifically. There is only one listen address configuration in the cluster's server-template. To effectively set the listen-address properly for each dynamic server in the cluster, a calculated macro must be used.
WebLogic Server provides the "${id}" macro which corresponds to the index number of the dynamic server in the cluster. This index starts at the numeral one ("1") and increments to the current managed server count for the cluster. This sequentially-numbered server ID macro can be used with the recommended host naming pattern to have the Listen address calculated for each Dynamic Server to listen on a specific network interface.

This approach is recommended for enterprise deployment environments where there is only one managed server per host per cluster and the cluster is expected to scale-out horizontally only.

To configure the server-template Listen Address using the ${id} macro:

1. Verify that the required entries in `/etc/hosts` are configured to the appropriate IP address for the intended machines.

   For example:
   For information about the requirements for name resolution, see Verifying IP Addresses and Host Names in DNS or Hosts File.

2. Browse to the Oracle WebLogic Server Administration console, and sign in with your administrative credentials.

   ```
   http://adminvhn:7001/console
   ```

3. **Lock & Edit** the domain.
4. Navigate to **Clusters** > **Server Templates**, and select the server template to be modified.
5. Set the Listen Address value to the appropriate abstracted listener hostname, with the variable assignment as written.

   For example:

   ```
   wsmpm-server-template Listen Address = OIMHOST${id}
   ```

6. Click **Save**.
7. Repeat from step 4 if additional server templates need to be modified.
8. Click **Activate Changes**.
9. Restart the servers that use the template, for the changes to be effective.

Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur. If you implement dynamic clusters, the configuration of the server template assigned to each newly added cluster should be verified and updated, otherwise, verify and update every statically-defined Managed Server for the newly added clusters.

> **Note:**
>
> This task is not required for Access Infrastructure.

This step is necessary to avoid potential issues when you perform remote deployments and for deployments that require the stage mode.

To update the directory paths for the Deployment Stage and Upload locations, complete the following steps:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the left navigation tree, expand **Domain**, and then **Environment**.
3. Click **Lock & Edit**.
4. Navigate to and edit the appropriate objects for your cluster type.
   a. For Static Clusters, navigate to **Servers** and click the name of the Managed Server you want to edit.
   b. For Dynamic Clusters, navigate to **Clusters** > **Server Templates**, and click on the name of the server template to be edited.
5. For each new Managed Server or Server Template to be edited:
   a. Click the **Configuration** tab, and then click the **Deployment** tab.
   b. Verify that the **Staging Directory Name** is set to the following:

      ```
      MSERVER_HOME/servers/server_or_template_name/stage
      ```

      Replace *MSERVER_HOME* with the full path for the *MSERVER_HOME* directory.

      If you use static clusters, update with the correct name of the Managed Server that you are editing.

      If you use dynamic clusters, leave the template name intact. For example: `/u02/oracle/config/domains/iamedg_domain/servers/XYZ-server-template/stage`
   c. Update the **Upload Directory Name** to the following value:

      ```
      ASERVER_HOME/servers/AdminServer/upload
      ```

      Replace *ASERVER_HOME* with the directory path for the *ASERVER_HOME* directory.
   d. Click **Save**.

   e. Return to the Summary of Servers or Summary of Server Templates screen as applicable.

6. Repeat the previous steps for each of the new managed servers or dynamic cluster server templates.
7. Navigate to and update the Upload Directory Name value for the AdminServer:
   a. Navigate to **Servers**, and select the AdminServer.
   b. Click the **Configuration** tab, and then click the **Deployment** Tab.
   c. Verify that the **Staging Directory Name** is set to the following absolute path:

      ```
      ASERVER_HOME/servers/AdminServer/stage
      ```

   d. Update the **Upload Directory Name** to the following absolute path:

      ```
      ASERVER_HOME/servers/AdminServer/upload
      ```

      Replace `ASERVER_HOME` with the directory path for the `ASERVER_HOME` directory.

   e. Click **Save**.
8. When you have modified all the appropriate objects, click **Activate Changes**.
9. Restart all Managed Servers for the changes to take effect.

> **Note:**
>
> If you continue directly with further domain configurations, a restart to enable the stage and upload directory changes is not strictly necessary at this time.

Setting the Front End Host and Port for a WebLogic Cluster

You must set the front-end HTTP host and port for the Oracle WebLogic Server cluster that hosts the Oracle Identity and Access Management servers. You can specify these values in the Configuration Wizard while you are specifying the properties of the domain.

However, when you add a SOA cluster as part of an Oracle Identity and Access Management enterprise deployment, Oracle recommends that you perform this task after you verify the SOA Managed Servers.

To set the frontend host and port from the Weblogic Server Administration Console:

1. Log in to the WebLogic Server Administration Console.
2. In the Change Center, click **Lock & Edit**.
3. In the Domain Structure panel, expand **Environment**, and click **Clusters**.
4. On the Clusters page, click the cluster that you want to modify, and then select the **HTTP** tab.
5. Use the information in Table 19-1 to add the required frontend hostname and port to each cluster.

Table 19-1 The Frontend Hostname and Port for Each Cluster

Name	Frontend Host	Frontend HTTP Port	Frontend HTTPs
OAM_Cluster	login.example.com		443

Table 19-1 (Cont.) The Frontend Hostname and Port for Each Cluster

Name	Frontend Host	Frontend HTTP Port	Frontend HTTPs
AMA_Cluster	iadadmin.example.com	80	
OIM_Cluster			
SOA_Cluster	igdinternal.example.com	7777	
WSM-PM_Cluster			

6. Click **Save**.
7. Click **Activate Changes**.
8. Restart the managed servers of the cluster.

Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer

It is important to understand how to enable SSL communication between the middle tier and the hardware load balancer.

> **Note:**
>
> The following steps are applicable if the hardware load balancer is configured with SSL and the front-end address of the system has been secured accordingly.

- When is SSL Communication Between the Middle Tier and Load Balancer Necessary?
- Generating Self-Signed Certificates Using the utils.CertGen Utility
- Creating an Identity Keystore Using the utils.ImportPrivateKey Utility
- Creating a Trust Keystore Using the Keytool Utility
- Importing the Load Balancer Certificate into the Truststore
- Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts
- Configuring WebLogic Servers to Use the Custom Keystores

When is SSL Communication Between the Middle Tier and Load Balancer Necessary?

In an enterprise deployment, there are scenarios where the software running on the middle tier must access the front-end SSL address of the hardware load balancer. In these scenarios, an appropriate SSL handshake must take place between the load balancer and the invoking servers. This handshake is not possible unless the Administration Server and Managed Servers on the middle tier are started by using the appropriate SSL configuration.

Generating Self-Signed Certificates Using the utils.CertGen Utility

This section describes the procedure to create self-signed certificates on OIMHOST1. Create certificates for every app-tier host by using the network name or alias of each host.

The directory where keystores and trust keystores are maintained must be on shared storage that is accessible from all nodes so that when the servers fail over (manually or with server migration), the appropriate certificates can be accessed from the failover node. Oracle recommends that you use central or shared stores for the certificates used for different purposes (for example, SSL set up for HTTP invocations). See the information on filesystem specifications for the KEYSTORE_HOME location provided in About the Recommended Directory Structure for an Enterprise Deployment.

For information on using trust CA certificates instead, see the information about configuring identity and trust in *Administering Security for Oracle WebLogic Server*.

About Passwords

The passwords used in this guide are used only as examples. Use secure passwords in a production environment. For example, use passwords that include both uppercase and lowercase characters as well as numbers.

To create self-signed certificates:

1. Temporarily, set up your environment by running the following script:

   ```
   . WL_HOME/server/bin/setWLSEnv.sh
   ```

   Note that there is a dot(.) and space( ) preceding the script name in order to source the shell script in the current shell.
2. Verify that the *CLASSPATH* environment variable is set:

   ```
   echo $CLASSPATH
   ```
3. Verify that the shared configuration directory folder has been created and mounted to shared storage correctly, as described in Preparing the File System for an Enterprise Deployment.

   For example, use the following command to verify that the shared configuration directory is available to each host:

   ```
   df -h | grep -B1 SHARED_CONFIG_DIR
   ```

   Replace *SHARED_CONFIG_DIR* with the actual path to your shared configuration directory.

   You can also do a listing of the directory to ensure that it is available to the host:

   ```
   ls -al SHARED_CONFIG_DIR
   ```
4. Create the keystore home folder structure if does not already exist.

   For example:

   ```
   cd SHARED_CONFIG_DIR
   mkdir keystores
   chown oracle:oinstall keystores
   chmod 750 keystores
   export KEYSTORE_HOME=SHARED_CONFIG_DIR/keystores
   ```
5. Change directory to the keystore home:

   ```
   cd KEYSTORE_HOME
   ```

6. Run the `utils.CertGen` tool to create the certificates for hostnames or aliases used by the managed servers and node managers, one per host.

   > **Note:**
   >
   > You must run the `utils.CertGen` tool to create certificates for all the other hosts that run the Manager Servers.

   Syntax:

   ```
   java utils.CertGen key_passphrase cert_file_name key_file_name [export | domestic]
   [hostname]
   ```

   Examples:

   ```
   java utils.CertGen password ADMINVHN.example.com_cert \
         ADMINVHN.example.com_key domestic ADMINVHN.example.com

   java utils.CertGen password OIMHOST1.example.com_cert \
         OIMHOST1.example.com_key domestic OIMHOST1.example.com
   ```

7. Repeat the above step for all the remaining hosts used in the system.
8. For Dynamic clusters, in addition to `ADMINVHN` and one certificate for each host, a certificate matching a wildcard URL should also be generated.

   For example:

   ```
   java utils.CertGen password WILDCARD.example.com_cert \
   WILDCARD.example.com_key domestic \*.example.com
   ```

Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

This section describes how to create an Identity Keystore on `OIMHOST1.example.com`.

In previous sections you have created certificates and keys that reside on shared storage. In this section, the certificate and private keys created earlier for all hosts and ADMINVHN are imported into a new Identity Store. Make sure that you use a different alias for each of the certificate and key pair imported.

> **Note:**
>
> The Identity Store is created (if none exists) when you import a certificate and the corresponding key into the Identity Store by using the `utils.ImportPrivateKey` utility.

1. Import the certificate and private key for ADMINVHN and OIMHOST1 into the Identity Store. Make sure that you use a different alias for each of the certificate and key pair imported.

   Syntax:

   ```
   java utils.ImportPrivateKey
         -certfile cert_file
         -keyfile private_key_file
   ```

```
      [-keyfilepass private_key_password]
      -keystore keystore
      -storepass storepass
      [-storetype storetype]
      -alias alias
      [-keypass keypass]
```

> **Note:**
>
> The default keystore_type is `jks`.

Examples:

```
java utils.ImportPrivateKey\
     -certfile KEYSTORE_HOME/ADMINVHN.example.com_cert.pem\
     -keyfile KEYSTORE_HOME/ADMINVHN.example.com_key.pem\
     -keyfilepass password\
     -keystore appIdentityKeyStore.jks\
     -storepass password\
     -alias ADMINVHN\
     -keypass password

java utils.ImportPrivateKey\
     -certfile KEYSTORE_HOME/OAMHOST1.example.com_cert.pem\
     -keyfile KEYSTORE_HOME/OAMHOST1.example.com_key.pem\
     -keyfilepass password\
     -keystore appIdentityKeyStore.jks\
     -storepass password\
     -alias OIMHOST1\
     -keypass password
```

2. Repeat the `java importPrivateKey` command for each of the remaining host-specific certificate and key pairs. (for example, for OAMHOST1, OAMHOST2).

> **Note:**
>
> Make sure to use a unique alias for each certificate and key pair imported.

3. For Dynamic clusters, import the wildcard certificate and private key pair by using the custom id alias of `WILDCARD`.

   Example:

```
${JAVA_HOME}/bin/java utils.ImportPrivateKey \
-certfile ${KEYSTORE_HOME}/WILDCARD.example.com_cert.pem \
-keyfile ${KEYSTORE_HOME}/WILDCARD.example.com_key.pem \
-keyfilepass password \
-keystore ${KEYSTORE_HOME}/appIdentityKeyStore.jks \
-storepass password \
-alias WILDCARD \
-keypass password
```

Creating a Trust Keystore Using the Keytool Utility

To create the Trust Keystore on OIMHOST1.example.com:

1. Copy the standard java keystore to create the new trust keystore since it already contains most of the root CA certificates needed.

   Oracle does not recommend modifying the standard Java trust key store directly. Copy the standard Java keystore CA certificates located under the `WL_HOME/server/lib` directory to the same directory as the certificates. For example:

   ```
   cp WL_HOME/server/lib/cacerts KEYSTORE_HOME/appTrustKeyStore.jks
   ```

2. Use the keytool utility to change the default password.

   The default password for the standard Java keystore is `changeit`. Oracle recommends that you always change the default password, as follows:

   ```
   keytool -storepasswd -new NewPassword -keystore TrustKeyStore -storepass
   Original_Password
   ```

   For example:

   ```
   keytool -storepasswd -new password -keystore appTrustKeyStore.jks -storepass changeit
   ```

3. Import the CA certificate into the `appTrustKeyStore` by using the keytool utility.

   The CA certificate `CertGenCA.der` is used to sign all certificates generated by the `utils.CertGen` tool and is located at `WL_HOME/server/lib` directory.

   Use the following syntax to import the certificate:

   ```
   keytool -import -v -noprompt -trustcacerts -alias AliasName -file CAFileLocation -
   keystore KeyStoreLocation -storepass KeyStore_Password
   ```

   For example:

   ```
   keytool -import -v -noprompt -trustcacerts -alias clientCACert -file WL_HOME/
   server/lib/CertGenCA.der -keystore appTrustKeyStore.jks -storepass password
   ```

Importing the Load Balancer Certificate into the Truststore

For the SSL handshake to act properly, the load balancer's certificate must be added to the WLS servers truststore. To add a load balancer’s certificate:

1. Access the site on SSL with a browser (this adds the server's certificate to the browser's repository).

2. Obtain the certificate from the load balancer. You can obtain the load balancer certificate using a browser such as Firefox. However, the easiest way to obtain the certificate is to use the `openssl` command. The syntax of the command is as follows:

   ```
   openssl s_client -connect LOADBALANCER -showcerts </dev/null 2>/dev/null|
   openssl x509 -outform PEM > SHARED_CONFIG_DIR/keystores/LOADBALANCER.pem
   ```

   For example:

   ```
   openssl s_client -connect prov.example.com:443 -showcerts </dev/null 2>/dev/
   null|openssl x509 -outform PEM > SHARED_CONFIG_DIR/keystores/
   prov.example.com.pem
   ```

3. Use the keytool to import the load balancer's certificate into the truststore:

   For example:

```
keytool -import -file SHARED_CONFIG_DIR/keystores/login.example.com -v -keystore
appTrustKeyStore.jks -alias aliasLogin -storepass password
keytool -import -file SHARED_CONFIG_DIR/keystores/prov.example.com.crt -v -keystore
appTrustKeyStore.jks -alias aliasProv -storepass password
```

4. Repeat this procedure for each SSL load balancer virtual host in your deployment.

> **Note:**
>
> The need to add the load balancer certificate to the WLS server truststore applies only to self-signed certificates. If the load balancer certificate is issued by a third-party CA, you have to import the public certificates of the root and the intermediate CAs into the truststore.

Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts

The `setUserOverridesLate.sh` script is supported by Oracle WebLogic Server as of this release and should be used to override default configurations set in the `setDomainEnv.sh` script that is invoked when starting the Administration Server and the Managed Servers in the domain. It is recommended not to edit the `setDomainEnv.sh` script as this script is regenerated during pack or unpack operations. Customizations to `setDomainEnv.sh` will be lost and require continuous maintenance. To ensure that each server accesses the updated trust store properly, edit the `setUserOverridesLate.sh` script in each of the domain home directories in the enterprise deployment. This file will also be maintained appropriately when using the pack or unpack commands.

1. Sign in to OIMHOST1 and open the following file with a text editor:

   ```
   IGD_ASERVER_HOME/bin/setUserOverridesLate.sh
   ```

2. Add commands to set the trustStore parameter on the `EXTRA_JAVA_PROPERTIES` variable with the correct path and filename for your custom trust store:

   For example:

   ```
   KEYSTORE_HOME/appTrustKeyStore.jks
   ```

> **Note:**
>
> All the values for `EXTRA_JAVA_PROPERTIES` must be on one line in the file, followed by the export command on a new line.
>
> ```
> echo ""
> echo "*****************************************************"
> echo "** Executing setUserOverrideLate.sh"
> echo "*****************************************************"
> echo ""
>
> #
> # Customize SSL Trust Store
> #
> EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES} -
> Djavax.net.ssl.trustStore=/u01/oracle/config/keystores/
> appTrustKeyStore.jks "
> export EXTRA_JAVA_PROPERTIES
>
> echo "EXTRA_JAVA_PROPERTIES=\"${EXTRA_JAVA_PROPERTIES}\""
> echo ""
> echo "*****************************************************"
> echo "** End of setUserOverrideLate.sh"
> echo "*****************************************************"
> echo ""
> ```

3. Copy the `*ASERVER_HOME*/bin/setUserOverridesLate.sh` file to the `*MSERVER_HOME*/bin` directory on OIMHOST1, OIMHOST2, OAMHOST1, and OAMHOST2.

> **Note:**
>
> The `setUserOverridesLate.sh` script must be used, not the `setUserOverrides.sh` script, otherwise some product components may override values set in `setUserOverrides.sh`. `setUserOverridesLate.sh` will also be propagated when using the pack and unpack tools.

Configuring WebLogic Servers to Use the Custom Keystores

Configure the WebLogic Servers to use the custom keystores by using the Oracle WebLogic Server Administration Console. Complete this procedure for the Administration Server and the Managed Servers that require access to the front-end LBR on SSL.

To configure the identity and trust keystores:

1. Log in to the Administration Console, and click **Lock & Edit**.
2. Navigate based on the Managed Server type:

   For configured Managed Servers:

   a. In the Domain Structure pane, expand **Environment** and select **Servers**.
   b. Click the name of the server for which you want to configure the identity and trust keystores.

For dynamic Managed Servers:

  a. In the Domain Structure pane, expand **Environment**, then **Clusters**, and then select **Server Templates**.

  b. Click the name of the appropriate server template for which you want to configure the identity and trust keystores.

3. Select **Configuration**, and then **Keystores**.
4. In the **Keystores** field, click **Change**, and select **Custom Identity and Custom Trust** method for storing and managing private keys and digital certificate pairs and trusted CA certificates, and click Save.
5. In the Identity section, define attributes for the identity keystore.
    - Custom Identity Keystore: Enter the fully qualified path to the identity keystore:

      ```
      KEYSTORE_HOME/appIdentityKeyStore.jks
      ```

    - Custom Identity Keystore Type: Leave this field blank, it defaults to JKS.
    - Custom Identity Keystore Passphrase: Enter the password Keystore_Password you provided in Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

      This attribute may be optional or required depending on the type of keystore. All keystores require the passphrase in order to write to the keystore. However, some keystores do not require the passphrase to read from the keystore. WebLogic Server reads only from the keystore, so whether or not you define this property depends on the requirements of the keystore.
6. In the Trust section, define properties for the trust keystore:
    - Custom Trust Keystore: Enter the fully qualified path to the trust keystore:

      ```
      KEYSTORE_HOME/appTrustKeyStore.jks
      ```

    - Custom Trust Keystore Type: Leave this field blank, it defaults to JKS.
    - Custom Trust Keystore Passphrase: The password you provided as the New_Password value in Creating a Trust Keystore Using the Keytool Utility.

      As mentioned in the previous step, this attribute may be optional or required depending on the type of keystore.
7. Click **Save**.
8. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.
9. Click **Lock & Edit**.
10. Select **Configuration**, then **SSL**.
11. Update the SSL Identity details as follows:

    a. In the **Private Key Alias** field, enter the alias value for the appropriate private key.

    - **With a Static Cluster**: Enter the alias that corresponds to the host the managed server listens on.
    - **With a Dynamic Cluster**: Enter the wildcard alias so any dynamic managed server can match any server.

    b. In the **Private Key Passphrase** and the **Confirm Private Key Passphrase** fields, enter the password for the keystore that you created in Creating an Identity Keystore Using the utils.ImportPrivateKey Utility.
12. Click **Save**.

13. If you are updating a server template SSL configuration for a dynamic cluster, perform these additional tasks:
    a. Click the **Advanced** link at the bottom of the SSL view.
    b. Select the **Custom Hostname Verifier** option from the HostName Verification menu.
    c. Set the Custom Hostname Verifier value to: `weblogic.security.utils.SSLWLSWildcardHostnameVerifier`.
    d. Click **Save**.
14. Click **Activate Changes** in the Administration Console's Change Center to make the changes take effect.
15. Restart the Administration Server.
16. Restart the Managed Servers where the keystore has been updated.

> **Note:**
>
> The fact that servers can be restarted by using the Administration Console and Node Manager is a good verification that the communication between Node Manager, Administration Server, and the managed servers is correct.

Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment

The persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence.

For example, the JMS subsystem stores persistent JMS messages and durable subscribers, and the JTA Transaction Log (TLOG) stores information about the committed transactions that are coordinated by the server but may not have been completed. The persistent store supports persistence to a file-based store or to a JDBC-enabled database. Persistent stores' high availability is provided by server or service migration. Server or service migration requires that all members of a WebLogic cluster have access to the same transaction and JMS persistent stores (regardless of whether the persistent store is file-based or database-based).

For an enterprise deployment, Oracle recommends using JDBC persistent stores for transaction logs (TLOGs) and JMS.

This section analyzes the benefits of using JDBC versus File persistent stores and explains the procedure for configuring the persistent stores in a supported database. If you want to use File persistent stores instead of JDBC stores, the procedure for configuring them is also explained in this section.

- Products and Components that use JMS Persistence Stores and TLOGs
- JDBC Persistent Stores vs. File Persistent Stores
- Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment
- Using File Persistent Stores for TLOGs and JMS in an Enterprise Deployment

Products and Components that use JMS Persistence Stores and TLOGs

Determining which installed FMW products and components utilize persistent stores can be done through the WebLogic Server Console in the Domain Structure navigation under ***DomainName*** > **Services** > **Persistent Stores**. The list indicates the name of the store, the store type (FileStore and JDBC), and the target of the store. The stores listed that pertain to MDS are outside the scope of this chapter and should not be considered.

These components (as applicable) use stores by default:

Component/Product	JMS Stores	TLOG Stores
B2B	Yes	Yes
BAM	Yes	Yes
BPM	Yes	Yes
ESS	No	No
HC	Yes	Yes
Insight	Yes	Yes
MFT	Yes	Yes
OSB	Yes	Yes
SOA	Yes	Yes
WSM	No	No

Component/Product	JMS Stores	TLOG Stores
OAM	No	No
OIM	Yes	Yes

JDBC Persistent Stores vs. File Persistent Stores

Oracle Fusion Middleware supports both database-based and file-based persistent stores for Oracle WebLogic Server transaction logs (TLOGs) and JMS. Before you decide on a persistent store strategy for your environment, consider the advantages and disadvantages of each approach.

> **Note:**
>
> Regardless of which storage method you choose, Oracle recommends that for transaction integrity and consistency, you use the same type of store for both JMS and TLOGs.

- About JDBC Persistent Stores for JMS and TLOGs
- Performance Considerations for TLOGs and JMS Persistent Stores

About JDBC Persistent Stores for JMS and TLOGs

When you store your TLOGs and JMS data in an Oracle database, you can take advantage of the replication and high availability features of the database. For example, you can use Oracle Data Guard to simplify cross-site synchronization. This is especially important if you are deploying Oracle Fusion Middleware in a disaster recovery configuration.

Storing TLOGs and JMS data in a database also means that you do not have to identity a specific shared storage location for this data. Note, however, that shared storage is still required for other aspects of an enterprise deployment. For example, it is necessary for Administration Server configuration (to support Administration Server failover), for deployment plans, and for adapter artifacts, such as the File and FTP Adapter control and processed files.

If you are storing TLOGs and JMS stores on a shared storage device, then you can protect this data by using the appropriate replication and backup strategy to guarantee zero data loss, and you potentially realize better system performance. However, the file system protection is always inferior to the protection provided by an Oracle Database.

For more information about the potential performance impact of using a database-based TLOGs and JMS store, see Performance Considerations for TLOGs and JMS Persistent Stores.

Performance Considerations for TLOGs and JMS Persistent Stores

One of the primary considerations when you select a storage method for Transaction Logs and JMS persistent stores is the potential impact on performance. This topic provides some guidelines and details to help you determine the performance impact of using JDBC persistent stores for TLOGs and JMS.

Performance Impact of Transaction Logs Versus JMS Stores

For transaction logs, the impact of using a JDBC store is relatively small, because the logs are very transient in nature. Typically, the effect is minimal when compared to other database operations in the system.

On the other hand, JMS database stores can have a higher impact on performance if the application is JMS intensive.

Factors that Affect Performance

There are multiple factors that can affect the performance of a system when it is using JMS DB stores for custom destinations. The main ones are:

- Custom destinations involved and their type
- Payloads being persisted
- Concurrency on the SOA system (producers on consumers for the destinations)

Depending on the effect of each one of the above, different settings can be configured in the following areas to improve performance:

- Type of data types used for the JMS table (using raw versus lobs)
- Segment definition for the JMS table (partitions at index and table level)

Impact of JMS Topics

If your system uses Topics intensively, then as concurrency increases, the performance degradation with an Oracle RAC database will increase more than for Queues. In tests conducted by Oracle with JMS, the average performance degradation for different payload sizes and different concurrency was less than 30% for Queues. For topics, the impact was more than 40%. Consider the importance of these destinations from the recovery perspective when deciding whether to use database stores.

Impact of Data Type and Payload Size

When you choose to use the RAW or SecureFiles LOB data type for the payloads, consider the size of the payload being persisted. For example, when payload sizes range between 100b and 20k, then the amount of database time required by SecureFiles LOB is slightly higher than for the RAW data type.

More specifically, when the payload size reach around 4k, then SecureFiles tend to require more database time. This is because 4k is where writes move out-of-row. At around 20k payload size, SecureFiles data starts being more efficient. When payload sizes increase to more than 20k, then the database time becomes worse for payloads set to the RAW data type.

One additional advantage for SecureFiles is that the database time incurred stabilizes with payload increases starting at 500k. In other words, at that point it is not relevant (for SecureFiles) whether the data is storing 500k, 1MB or 2MB payloads, because the write is asynchronized, and the contention is the same in all cases.

The effect of concurrency (producers and consumers) on the queue’s throughput is similar for both RAW and SecureFiles until the payload sizes reach 50K. For small payloads, the effect on varying concurrency is practically the same, with slightly better scalability for RAW. Scalability is better for SecureFiles when the payloads are above 50k.

Impact of Concurrency, Worker Threads, and Database Partioning

Concurrency and worker threads defined for the persistent store can cause contention in the RAC database at the index and global cache level. Using a reverse index when enabling multiple worker threads in one single server or using multiple Oracle WebLogic Server clusters can improve things. However, if the Oracle Database partitioning option is available, then global hash partition for indexes should be used instead. This reduces the contention on the index and the global cache buffer waits, which in turn improves the response time of the application. Partitioning works well in all cases, some of which will not see significant improvements with a reverse index.

Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment

This section explains the guidelines to use JDBC persistent stores for transaction logs (TLOGs) and JMS. It also explains the procedures to configure the persistent stores in a supported database.

- Recommendations for TLOGs and JMS Datasource Consolidation
  To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.
- Roadmap for Configuring a JDBC Persistent Store for TLOGs
  The following topics describe how to configure a database-based persistent store for transaction logs.
- Roadmap for Configuring a JDBC Persistent Store for JMS
  The following topics describe how to configure a database-based persistent store for JMS.
- Creating a User and Tablespace for TLOGs
  Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.
- Creating a User and Tablespace for JMS
  Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.
- Creating GridLink Data Sources for TLOGs and JMS Stores
  Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.

- Assigning the TLOGs JDBC Store to the Managed Servers
  If you are going to accomplish data source consolidation, you will reuse the `<PREFIX>_WLS` tablespace and `WLSSchemaDatasource` for the TLOG persistent store. Otherwise, ensure that you create the tablespace and user in the database, and you have created the datasource before you assign the TLOG store to each of the required Managed Servers.
- Creating a JDBC JMS Store
  After you create the JMS persistent store user and table space in the database, and after you create the data source for the JMS persistent store, you can then use the Administration Console to create the store.
- Assigning the JMS JDBC store to the JMS Servers
  After you create the JMS tablespace and user in the database, create the JMS datasource, and create the JDBC store, then you can assign the JMS persistence store to each of the required JMS Servers.
- Creating the Required Tables for the JMS JDBC Store
  The final step in using a JDBC persistent store for JMS is to create the required JDBC store tables. Perform this task before you restart the Managed Servers in the domain.

Recommendations for TLOGs and JMS Datasource Consolidation

To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.

Oracle recommends you to reuse the `WLSSchemaDatasource` as is for TLOGs and JMS persistent stores under non-high workloads and consider increasing the `WLSSchemaDatasource` pool size. Reuse of datasource forces to use the same schema and tablespaces, and so the `PREFIX_WLS_RUNTIME` schema in the `PREFIX_WLS` tablespace is used for both TLOGs and JMS messages.

High stress (related with high JMS activity, for example) and contention in the datasource can cause stability and performance problems. For example:

- High contention in the DataSource can cause persistent stores to fail if no connections are available in the pool to persist JMS messages.
- High Contention in the DataSource can cause issues in transactions if no connections are available in the pool to update transaction logs.

For these cases, use a separate datasource for TLOGs and stores and a separate datasource for the different stores. You can still reuse the `PREFIX_WLS_RUNTIME` schema but configure separate custom datasources to the same schema to solve the contention issue.

Roadmap for Configuring a JDBC Persistent Store for TLOGs

The following topics describe how to configure a database-based persistent store for transaction logs.

1. Creating a User and Tablespace for TLOGs
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Assigning the TLOGs JDBC Store to the Managed Servers

> **Note:**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Roadmap for Configuring a JDBC Persistent Store for JMS

The following topics describe how to configure a database-based persistent store for JMS.

1. Creating a User and Tablespace for JMS
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Creating a JDBC JMS Store
4. Assigning the JMS JDBC store to the JMS Servers
5. Creating the Required Tables for the JMS JDBC Store

> **Note:**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Creating a User and Tablespace for TLOGs

Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.

1. Create a tablespace called `tlogs`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace tlogs
          logging datafile 'path-to-data-file-or-+asmvolume'
          size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `TLOGS` and assign to it the `tlogs` tablespace.

   For example:

   ```
   SQL> create user TLOGS identified by password;

   SQL> grant create table to TLOGS;

   SQL> grant create session to TLOGS;

   SQL> alter user TLOGS default tablespace tlogs;

   SQL> alter user TLOGS quota unlimited on tlogs;
   ```

Creating a User and Tablespace for JMS

Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.

1. Create a tablespace called `jms`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace jms
           logging datafile 'path-to-data-file-or-+asmvolume'
           size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `JMS` and assign to it the `jms` tablespace.

   For example:

   ```
   SQL> create user JMS identified by password;

   SQL> grant create table to JMS;

   SQL> grant create session to JMS;

   SQL> alter user JMS default tablespace jms;

   SQL> alter user JMS quota unlimited on jms;
   ```

Creating GridLink Data Sources for TLOGs and JMS Stores

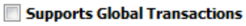
Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.

For an enterprise deployment, you should use GridLink data sources for your TLOGs and JMS stores. To create a GridLink data source:

1. Sign in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Data Sources**.
4. On the Summary of Data Sources page, click **New** and select **GridLink Data Source**, and enter the following:
   - Enter a logical name for the data source in the **Name** field.

     For the TLOGs store, enter TLOG; for the JMS store, enter JMS.
   - Enter a name for **JNDI**.

     For the TLOGs store, enter `jdbc/tlogs`; for the JMS store, enter `jdbc/jms`.
   - For the Database Driver, select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
   - Click **Next**.
5. In the Transaction Options page, clear the **Supports Global Transactions** check box, and then click **Next**.



□ Supports Global Transactions

6. In the GridLink Data Source Connection Properties Options screen, select **Enter individual listener information** and click **Next**.

7. Enter the following connection properties:

   - **Service Name**: Enter the service name of the database with lowercase characters. For a GridLink data source, you must enter the Oracle RAC service name. For example:

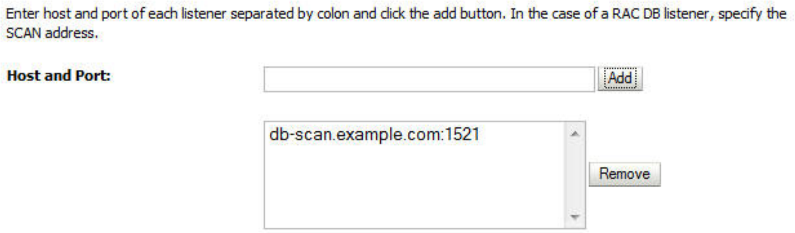
     ```
     iamedg.example.com
     ```

   - **Host Name and Port**: Enter the SCAN address and port for the RAC database, separated by a colon. For example:

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

     Click **Add** to add the host name and port to the list box below the field.

     Figure 19-1 Adding Host Name and Port Details for the RAC Database

     Enter host and port of each listener separated by colon and click the add button. In the case of a RAC DB listener, specify the SCAN address.

     Host and Port:

     db-scan.example.com:1521

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

     ```
     SQL>show parameter remote_listener;

     NAME                    TYPE        VALUE

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

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

     > **Note:**
     >
     > For Oracle Database 11*g* Release 1 (11.1), use the virtual IP and port of each database instance listener, for example:
     >
     > ```
     > dbhost1-vip.example.com (port 1521)
     > ```
     >
     > and
     >
     > ```
     > dbhost2-vip.example.com (1521)
     > ```

   - **Database User Name**: Enter the following:

     For the TLOGs store, enter `TLOGS`; for the JMS persistent store, enter `JMS`.

   - **Password**: Enter the password that you used when you created the user in the database.

- **Confirm Password**: Enter the password again and click **Next**.

8. On the Test GridLink Database Connection page, review the connection parameters and click **Test All Listeners**.

   Here is an example of a successful connection notification:

   ```
   Connection test for
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=db-
   scan.example.com)
   (PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=iamedg.example.com))) succeeded.
   ```

   Click **Next**.

9. In the ONS Client Configuration page, do the following:

   - Select **FAN Enabled** to subscribe to and process Oracle FAN events.
   - Enter the SCAN address: ONS remote port for the RAC database and the ONS remote port as reported by the database (see the following example) and click **Add**:

     ```
     [orcl@db-scan1 ~]$ srvctl config nodeapps -s

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

   - Click **Next**.

   > **Note:**
   >
   > For Oracle Database 11*g* Release 1 (11.1), use the hostname and port of each database's ONS service, for example:
   >
   > ```
   > custdbhost1.example.com (port 6200)
   > ```
   >
   > and
   >
   > ```
   > custdbhost2.example.com (6200)
   > ```

10. On the Test ONS Client Configuration page, review the connection parameters and click **Test All ONS Nodes**.

    Here is an example of a successful connection notification:

    ```
    Connection test for db-scan.example.com:6200 succeeded.
    ```

    Click **Next**.

11. In the Select Targets page, select the cluster that is using the persistent store, and then select **All Servers in the cluster**.

12. Click **Finish**.

13. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

14. Repeat step 4 through step 13 to create the GridLink Data Source for JMS File Stores.

Assigning the TLOGs JDBC Store to the Managed Servers

If you are going to accomplish data source consolidation, you will reuse the `<PREFIX>_WLS` tablespace and `WLSSchemaDatasource` for the TLOG persistent store. Otherwise, ensure that you create the tablespace and user in the database, and you have created the datasource before you assign the TLOG store to each of the required Managed Servers.

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Change Center**, click **Lock and Edit**.
3. To configure the TLOG of a Managed Server, in the Domain Structure tree:
   a. **For static clusters**: expand **Environment**, then **Servers**, and then click the name of the Managed Server.
   b. **For dynamic cluster**: expand **Environment**, then **Cluster**, and **Server Templates**. Click the name of the server template.
4. Select the **Configuration** > **Services** tab.
5. Under **Transaction Log Store**, select **JDBC** from the **Type** menu.
6. From the **Data Source** menu, select `WLSSchemaDatasource` to accomplish data source consolidation. The `<PREFIX>_WLS` tablespace will be used for TLOGs.
7. In the **Prefix Name** field, specify a prefix name to form a unique JDBC TLOG store name for each configured JDBC TLOG store
8. Click **Save**.
9. Repeat steps 3 to 7 for each additional managed server or server template.
10. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Creating a JDBC JMS Store

After you create the JMS persistent store user and table space in the database, and after you create the data source for the JMS persistent store, you can then use the Administration Console to create the store.

1. Log in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Persistent Store**.
4. Click **New**, and then click **JDBC Store**.
5. Enter a persistent store name that easily relates it to the pertaining JMS servers that is using it.

   > **Note:**
   >
   > The length of the prefix name must not exceed 30 characters for DB versions that are below 12.2.x.x.x.

6. To accomplish data source consolidation, select `WLSSchemaDatasource`. The `<PREFIX>_WLS` tablespace will be used for JMS persistent stores.
7. Target the store to the entity that hosts the JTA services.

   In the static cluster case, with a server that uses service migration, the entity is the migratable target to which the JMS server belongs.

In the case of a dynamic cluster, target to the cluster itself.

For more information about using dynamic clusters, see Simplified JMS Configuration and High Availability Enhancements in *Administering JMS Resources for Oracle WebLogic Server*.

8. Repeat steps 3 through 7 for each additional JMS server in the cluster.
9. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Assigning the JMS JDBC store to the JMS Servers

After you create the JMS tablespace and user in the database, create the JMS datasource, and create the JDBC store, then you can assign the JMS persistence store to each of the required JMS Servers.

To assign the JMS persistence store to the JMS servers:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Change Center**, click **Lock and Edit**.
3. In the Domain Structure tree, expand **Services**, then **Messaging**, and then **JMS Servers**.
4. Click the name of the JMS Server that you want to use the persistent store.
5. From the **Persistent Store** menu, select the JMS persistent store you created earlier.
6. Click **Save**.
7. Repeat steps 3 to 6 for each of the additional JMS Servers in the cluster.
8. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Creating the Required Tables for the JMS JDBC Store

The final step in using a JDBC persistent store for JMS is to create the required JDBC store tables. Perform this task before you restart the Managed Servers in the domain.

1. Review the information in Performance Considerations for TLOGs and JMS Persistent Stores, and decide which table features are appropriate for your environment.

   There are three Oracle DB schema definitions provided in this release and were extracted for review in the previous step. The basic definition includes the RAW data type without any partition for indexes. The second uses the blob data type, and the third uses the blob data type and secure files.

2. Create a domain-specific well-named folder structure for the custom `DDL` file on shared storage. The *ORACLE_RUNTIME* shared volume is recommended so it is available to all servers.

   Example:

   ```
   mkdir -p ORACLE_RUNTIME/domain_name/ddl
   ```

3. Create a `jms_custom.ddl` file in new shared `ddl` folder based on your requirements analysis.

For example, to implement an optimized schema definition that uses both secure files and hash partitioning, create the `jms_custom.ddl` file with the following content:

```
CREATE TABLE $TABLE (
  id     int  not null,
  type   int  not null,
  handle int  not null,
  record blob not null,
PRIMARY KEY (ID) USING INDEX GLOBAL PARTITION BY HASH (ID) PARTITIONS 8)
LOB (RECORD) STORE AS SECUREFILE (ENABLE STORAGE IN ROW);
```

This example can be compared to the default schema definition for JMS stores, where the RAW data type is used without any partitions for indexes.

Note that the number of partitions should be a power of two. This ensures that each partition is of similar size. The recommended number of partitions varies depending on the expected table or index growth. You should have your database administrator (DBA) analyze the growth of the tables over time and adjust the tables accordingly. See Partitioning Concepts in *Database VLDB and Partitioning Guide*.

4. Use the Administration Console to edit the existing JDBC Store you created earlier; create the table that is used for the JMS data:
   a. Login in to the Oracle WebLogic Server Administration Console.
   b. In the **Change Center**, click **Lock and Edit**.
   c. In the Domain Structure tree, expand **Services**, then **Persistent Stores**.
   d. Click the persistent store you created earlier.
   e. Under the **Advanced** options, enter `*ORACLE_RUNTIME*/*domain_name*/ddl/jms_custom.ddl` in the **Create Table from DDL File** field.
   f. Click **Save**.
   g. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.
5. Restart the Managed Servers.

Using File Persistent Stores for TLOGs and JMS in an Enterprise Deployment

This section explains the procedures to configure TLOGs and JMS File persistent stores in a shared folder.

- Configuring TLOGs File Persistent Store in a Shared Folder
- Configuring JMS File Persistent Store in a Shared Folder

Configuring TLOGs File Persistent Store in a Shared Folder

Oracle WebLogic Server uses the transaction logs to recover from system crashes or network failures.

- Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster
- Configuring TLOGs File Persistent Store in a Shared Folder with a Dynamic Cluster
- Validating the Location and Creation of the Transaction Logs

Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster

To set the location for the default persistence stores for each managed server in a static cluster, complete the following steps:

1. Log into the Oracle WebLogic Server Administration console:

   ```
   ADMINVHN:7001/console
   ```

   > **Note:**
   >
   > If you have already configured web tier, use `http://admin.example.com/console`.

2. In the Change Center section, click **Lock & Edit**.
3. For each of the Managed Servers in the cluster:
   a. In the Domain Structure window, expand the **Environment** node, and then click the **Servers** node.

      The Summary of Servers page appears.

   b. Click the name of the server (represented as a hyperlink) in the **Name** column of the table.

      The settings page for the selected server appears and defaults to the Configuration tab.

   c. On the **Configuration** tab, click the **Services** tab.

   d. In the Default Store section of the page, enter the path to the folder where the default persistent stores stores its data files.

      For the enterprise deployment, use the *ORACLE_RUNTIME* directory location. This subdirectory serves as the central, shared location for transaction logs for the cluster. See File System and Directory Variables Used in This Guide.

      For example:

      ```
      ORACLE_RUNTIME/domain_name/cluster_name/tlogs
      ```

      In this example, replace *ORACLE_RUNTIME* with the value of the variable for your environment. Replace *domain_name* with the name you assigned to the domain. Replace *cluster_name* with the name of the cluster you just created.

   e. Click **Save**.
4. Complete step 3 for all servers in the SOA_Cluster.
5. Click **Activate Changes**.

> **Note:**
>
> You validate the location and the creation of the transaction logs later in the configuration procedure.

Configuring TLOGs File Persistent Store in a Shared Folder with a Dynamic Cluster

To set the location for the default persistence stores for a dynamic cluster, update the server template:

1. Log into the Oracle WebLogic Server Administration Console:

   ```
   ADMINVHN:7001/console
   ```

   > **Note:**
   >
   > If you have already configured web tier, use `http://admin.example.com/console`.

2. In the Change Center section, click **Lock & Edit**.
3. Navigate to the server template for the cluster:
   a. In the Domain Structure window, expand the **Environment and Clusters** nodes, and then click the **Server Templates** node.

      The Summary of Server Templates page appears.

   b. Click the name of the server template (represented as a hyperlink) in the **Name** column of the table.

      The settings page for the selected server template appears and defaults to the **Configuration** tab.

   c. On the **Configuration** tab, click the **Services** tab.
   d. In the Default Store section of the page, enter the path to the folder where the default persistent stores stores its data files.

      For the enterprise deployment, use the *ORACLE_RUNTIME* directory location. This subdirectory serves as the central, shared location for transaction logs for the cluster. See File System and Directory Variables Used in This Guide.

      For example:

      ```
      ORACLE_RUNTIME/domain_name/cluster_name/tlogs
      ```

      In this example, replace *ORACLE_RUNTIME* with the value of the variable for your environment. Replace *domain_name* with the name that you assigned to the domain. Replace *cluster_name* with the name of the cluster you just created.

   e. Click **Save**.
4. Click **Activate Changes**.

> **Note:**
>
> You validate the location and the creation of the transaction logs later in the configuration procedure.

Validating the Location and Creation of the Transaction Logs

After the WLS_SERVER_TYPE1 and WLS_SERVER_TYPE2 managed Servers are up and running, verify that the transaction log directory and transaction logs are created as expected, based on the steps that you performed in Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster and Configuring TLOGs File Persistent Store in a Shared Folder with a Dynamic Cluster:

```
ORACLE_RUNTIME/domain_name/OSB_Cluster/tlogs
```

- `_WLS_WLS_SERVER_TYPE1000000.DAT`
- `_WLS_WLS_SERVER_TYPE2000000.DAT`

Configuring JMS File Persistent Store in a Shared Folder

If you have already configured and extended your domain, the JMS Persistent Files are already configured in a shared location. If you need to change any other persistent store file to the shared folder, perform the following steps:

1. Log in to the Oracle WebLogic Server Administration Console.
2. Navigate to **Domain > Services > Persistent Store** and click the name of the persistent store that you want to move to the shared folder.

   The **Configuration: General** tab is displayed.
3. Change the directory to `ORACLE_RUNTIME/domain_name/soa_cluster/jms`.
4. Click **Save**.
5. Click **Activate Changes**.

About JDBC Persistent Stores for Web Services

By default, web services use the WebLogic Server default persistent store for persistence. This store provides high-performance storage solution for web services.

The default web service persistence store is used by the following advanced features:

- Reliable Messaging
- Make Connection
- SecureConversation
- Message buffering

You also have the option to use a JDBC persistence store in your WebLogic Server web service, instead of the default store. For information about web service persistence, see Managing Web Service Persistence.

Performing Backups and Recoveries for an Enterprise Deployment

It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle Identity and Access Management enterprise deployment.

> **Note:**
>
> Some of the static and runtime artifacts listed in this section are hosted from Network Attached Storage (NAS). If possible, backup and recover these volumes from the NAS filer directly rather than from the application servers.

For general information about backing up and recovering Oracle Fusion Middleware products, see the following sections in *Administering Oracle Fusion Middleware*:

- Backing Up Your Environment
- Recovering Your Environment

Table 19-2 lists the static artifacts to back up in a typical Oracle Identity and Access Management enterprise deployment.

Table 19-2 Static Artifacts to Back Up in the Oracle Identity and Access Management Enterprise Deployment

Type	Host	Tier
Database Oracle home	DBHOST1 and DBHOST2	Data Tier
Oracle Fusion Middleware Oracle home	WEBHOST1 and WEBHOST2	Web Tier
Oracle Fusion Middleware Oracle home	OIMHOST1 and OIMHOST2 (or NAS Filer)	Application Tier
Installation-related files	WEBHOST1, WEHOST2, and shared storage	N/A

Table 19-3 lists the runtime artifacts to back up in a typical Oracle Identity and Access Management enterprise deployment.

Table 19-3 Run-Time Artifacts to Back Up in the Oracle Identity and Access Management Enterprise Deployment

Type	Host	Tier
Administration Server domain home (ASERVER_HOME)	OIMHOST1 (or NAS Filer)	Application Tier
Application home (APPLICATION_HOME)	OIMHOST1 (or NAS Filer)	Application Tier
Oracle RAC databases	DBHOST1 and DBHOST2	Data Tier
Scripts and Customizations	Per host	Application Tier
Deployment Plan home (DEPLOY_PLAN_HOME)	OIMHOST1 (or NAS Filer)	Application Tier
OHS Configuration directory	WEBHOST1 and WEBHOST2	Web Tier

Configuration and Management Tasks for an Oracle Identity and Access Management Enterprise Deployment

These are some of the key configuration and management tasks that you likely need to perform on an Oracle Identity and Access Management enterprise deployment.

- Using Shared Storage for Deployment Plans and SOA Infrastructure Applications Updates
- Managing the JMS Messages in a SOA Server

Using Shared Storage for Deployment Plans and SOA Infrastructure Applications Updates

When you redeploy a SOA infrastructure application or resource adapter within the SOA cluster, the deployment plan along with the application bits should be accessible to all servers in the cluster.

SOA applications and resource adapters are installed using nostage deployment mode. Because the administration sever does not copy the archive files from their source location when the nostage deployment mode is selected, each server must be able to access the same deployment plan.

To ensure deployment plan location is available to all servers in the domain, use the Deployment Plan home location described in File System and Directory Variables Used in This Guide and represented by the *DEPLOY_PLAN_HOME* variable in the *Enterprise Deployment Workbook*.

Managing the JMS Messages in a SOA Server

There are several procedures to manage JMS messages in a SOA server. You may need to perform these procedures in some scenarios, for example, to preserve the messages during a scale-in operation.

This section explains some of these procedures in detail.

- Draining the JMS Messages from a SOA Server

Draining the JMS Messages from a SOA Server

The process of draining the JMS messages helps you clear out the messages from a particular WebLogic server. A basic approach to drain stores consists of stopping the message production in the appropriate JMS Servers and allowing the applications to consume the messages.

This procedure, however, is application dependent, and could take an unpredictable amount of time. As an alternative, general instructions are provided here for saving the current messages from their current JMS destinations and, when/if required, importing them into a different server.

The draining procedure is useful in scale-in/down scenarios, where the size of the cluster is reduced by removing one or more servers. You can ensure that no messages are lost by draining the messages from the server that you delete, and then importing them into another server in the cluster.

You can also use this procedure in some disaster recovery maintenance scenarios, when the servers are started in a secondary location by using an Snapshot Standby database. In this case, you may need to drain the messages from the domain before starting it in the secondary location to avoid their consumption in the standby domain when you start the domain (otherwise, duplicate executions could take place). You cannot import messages in this scenario.

To drain the JMS messages from a server, perform the following steps:

1. Stop a new workload by pausing production for the JMS Server. You must do this activity for each JMS Server of the server that is affected in the operation:
   a. Navigate to the WebLogic Console and click **Environment** > **Services**> **JMS Server** ><*JMS Server name*>> **Control**.
   b. Select the *JMS Server* of the server that you want to delete.
   c. Click **Production**, and then click **Pause**.
2. Drain the messages from the destinations. To drain the JMS messages, you can let applications consume the pending messages. However, this task is application dependent and may take time. Hence, Oracle recommends you to export the messages of each destination. Verify which destinations have messages:
   a. Navigate to the WebLogic Console and click **Environment** > **Services**> **JMS Server**> **Monitoring** > **Active Destination**.
   b. Look whether the destination members of the server that you want to delete have current messages. Identify the destination name and its JMS Module.
   c. Repeat this activity for each JMS Server that is running in the server that you want to delete.
   - **Drain messages from queues**: For those queue destinations that have current messages:
     a. Navigate to the WebLogic Console and click **Environment** > **Services** > **JMS Module** > <*JMS module name*> > <*destination name*>.
     b. Click **Monitoring**.
     c. Select the queue corresponding with the server that you want to delete and click **Show Messages**.
     d. Select **Export** > **Export All** and export the messages to a file. Make a note of the file name for later use
     e. Delete the exported messages by using the **Delete All** option. This step is important to avoid message duplications.
   - **Drain messages from topics**

     Oracle recommends you to drain and import messages from topics only if they have a critical business impact. See Table 19-4 for details about the purpose and business impact for each topic. Only the loss of messages in the topic **dist_EDNTopic_auto**, used by EDN, has a business impact.

Table 19-4 Details of the Purpose and Business Impact for Each Topic of a Component

Component	JMS Module	JMS Topic Name	Purpose	Business Impact of Message Loss
SOA	SOAJMSModule	dist_B2BBroadcastTopic_auto	Used by B2B, messages are meant to be consumed immediately.	No impact.
SOA	SOAJMSModule	dist_EDNTopic_auto	Used for EDN, contains event messages for applications.	Business impact. Applications that consume these EDN event messages will lose them.
SOA	SOAJMSModule	dist_TenantTopic_auto	No longer used.	No impact.
SOA	SOAJMSModule	dist_XmlSchemaChangeNotificationTopic_auto	No longer used.	No impact.

Follow these steps drain messages from the topics:

a. Navigate to the WebLogic Console and click **Environment** > **Services** > **JMS Module** > <*JMS module name*> > <topic name>.

b. Click **Monitoring**, and then click **Durable Subscribers**.

c. Select the topic corresponding to the server that you want to delete and click **Apply**. The page displays the subscriptions only for the selected member topic.

d. Select the Durable Subscriber that has current messages and click**Show Messages**.

e. Click **Export** > **Export All** and export the messages to a file. Make a note of the file name for later use.

f. Delete the exported messages from the subscriber by clicking **Delete** > **Delete All**. This step is important to avoid message duplications.

g. Repeat the export process for any subscriber in the topic that has current messages.

Considerations for Cross-Component Wiring

Cross-Component Wiring (CCW) enables the FMW components to publish and bind to some of the services available in a WLS domain, by using specific APIs.

CCW performs a bind of the wiring information only during the Configuration Wizard session or when manually forced by the WLS domain Administrator. When you add a Weblogic Server to a cluster (in a scale out and scale up operation in a static or dynamic cluster), although the new server publishes its services, all the clients that use the service are not automatically updated and bound to the new service provider. The update does not happen because the existing servers that are already bound to a CCW table, do not automatically *know* about the new member that joins the cluster. It is the same case with ESS and WSMPM when they provide their services to SOA: both publish their service to the service table dynamically, but SOA servers do not know about these updates unless a bind is forced again.

> **Note:**
>
> There is an additional cross-component wiring information similar to the one used by the OHS configuration, which is not affected by this wiring because of the proxy plug-in behavior. For more information, see the following sections:
>
> - Wiring Components to Work Together in *Administering Oracle Fusion Middleware*.
> - Oracle-Developed Modules for Oracle HTTP Server in *Administering Oracle HTTP Server*

- Cross-Component Wiring for WSMPM and ESS
  The cross-component wiring t3 information is used by WSMPM and ESS to obtain the list of severs to be used in a JNDI invocation URL.
- Using the cluster_name Syntax with WSMPM
  This procedure makes WSMPM use a t3 syntax that accounts for servers being added or removed from the WSMPM cluster without having to reupdate the CCW information.

Cross-Component Wiring for WSMPM and ESS

The cross-component wiring t3 information is used by WSMPM and ESS to obtain the list of severs to be used in a JNDI invocation URL.

The CCW t3 information limits the impact of the lack of dynamic updates. When the invocation is done, the JNDI URL is used to obtain the RMI stubs with the list of members in the cluster. The JNDI URL does not need to contain the entire list of servers. The RMI stubs contain the list of all the servers in the cluster at any given time, and are used to load balance requests across all of them. Therefore, without a bind, the servers that are added to the cluster are used even if not present in the bind URL. The only drawback is that at least one of the original servers provided in the first CCW bind must be up to keep the system working when the cluster expands or shrinks. To avoid this issue, you can use the *cluster name* syntax in the service table instead of using the static list of members.

The cluster name syntax is as follows:

```
cluster:t3://cluster_name
```

When you use `cluster:t3://cluster_name`, the CCW invocation fetches the complete list of members in the cluster at any given time, thus avoiding any dependencies on the initial servers and accounting for every member that is alive in the cluster then.

Using the cluster_name Syntax with WSMPM

This procedure makes WSMPM use a t3 syntax that accounts for servers being added or removed from the WSMPM cluster without having to reupdate the CCW information.

The CCW t3 information is configured to use the cluster syntax by default. You only need to verify that the cluster syntax is used and edit, if required.

1. Sign-in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_iam`.

2. From the WebLogic Domain drop-down menu, select **Cross component Wiring- Service Tables**.
3. Select the **OWSM Policy Manager urn:oracle:fmw.owsm-pm:t3** row.
4. Verify that the cluster syntax is used. If not, click **Edit** and update the t3 and t3s values with the cluster name syntax.
5. Click **OK**.
6. From the WebLogic Domain drop-down menu, select **Cross component Wiring - Components**.
7. Select **OWSM Agent**.
8. In the Client Configuration section, select the **owsm-pm-connection-t3** row and click **Bind**.
9. Click **OK**.

> **Note:**
>
> The wiring table is updated with each cluster scale out or scale up, but it does not replace the cluster syntax until a manual rebind is used. Hence, it withstands all updates (additions and removals) in the lifecycle of the cluster.

Starting and Stopping Servers in Dynamic Clusters

You can start and stop server instances in dynamic clusters by using the same methods you use to start and stop server instances in configured static clusters.

Methods to start and stop server instances in configured clusters:

- WebLogic Server Administration Console
- Fusion Middleware Control
- WLST start and shutdown commands
- Node Manager
- Start scripts

Depending on which startup method you choose and the tasks you have already performed, you may have to follow several other procedures before you can start server instances. See Starting and Stopping Servers section in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

> **Note:**
>
> Before you begin, ensure that WebLogic Server is installed on all hosts where you want to run your server instances. If you want to use Node Manager to start and stop your server instances, then you must also run Node Manager on these hosts.

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 number of dynamic server instances available are based on the configured maximum specified in the server template for a given dynamic cluster. Transient changes in capacity requirements can be easily met by starting or stopping some of the available managed servers within the cluster, keeping in mind that a minimum of two or three are required to maintain high-availability.

If you need additional server capacity on top of the number of server instances you originally specified, you can 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.

You can also use the WLST `scaleUp` and `scaleDown` commands to manage your dynamic cluster. To increase the number of dynamic servers in the dynamic cluster, use the `scaleUp` command and enable the `updateConfiguration` argument. WLST will increase the maximum size of the cluster by the specified number of servers and start the server instances.

The `scaleUp` command increases the number of running servers for the specified dynamic cluster. The non-running server instance with the lowest server ID starts first, followed by the next highest non-running server ID, until the specified number of server instances is started.

You can start one, all, or any number of server instances in the dynamic cluster by specifying the desired number with the `numServers` argument in the `scaleUp` command. If all available server instances are already running, the `scaleUp` command increases the size of the cluster to the minimum number of requested server instances before starting the specified number of servers.

To decrease the maximum size of the dynamic cluster, use the scaleDown command and enable the `updateConfiguration` argument. WLST will gracefully shut down the specified number of running server instances and remove them from the dynamic cluster. See scaleUp and scaleDown in *WLST Command Reference for WebLogic Server*. The `scaleDown` command gracefully shuts down the specified number of running servers. The server instance with the highest server ID shuts down first, followed by the next highest ID, until the specified number of server instances is shut down.

Note:

You can only use the WLST `scaleUp` and `scaleDown` commands with dynamic server instances. In a mixed cluster, containing both manually configured and dynamic server instances, the `scaleUp` and `scaleDown` commands ignore the configured servers. You must manually start and stop configured server instances in a mixed cluster.

For example, a cluster contains two running dynamic servers and two non-running configured servers. If you use the `scaleUp` command, WLST adds one additional dynamic server instance to your cluster and starts the dynamic server.

The WLST `scaleUp` and `scaleDown` commands provide ways to manually scale your dynamic cluster. For automatic scaling, you can configure elasticity for your dynamic cluster. Elasticity enables a dynamic cluster to perform scaling and re-provisioning operations automatically in response to demand or on a calendar based schedule. WebLogic Server provides elasticity for dynamic clusters through the Policies and Actions system of the WebLogic Diagnostic Framework (WLDF). See Configuring Elasticity in Dynamic Clusters for Oracle WebLogic Server.

20 Using Whole Server Migration and Service Migration in an Enterprise Deployment

The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. The following sections explain how these features can be used in an Oracle Fusion Middleware enterprise topology.

- About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment
  Oracle WebLogic Server provides a migration framework that is an integral part of any highly available environment. The following sections provide more information about how this framework can be used effectively in an enterprise deployment.
- Creating a GridLink Data Source for Leasing
  Whole Server Migration and Automatic Service Migration require a data source for the leasing table, which is a tablespace created automatically as part of the Oracle WebLogic Server schemas by the Repository Creation Utility (RCU).
- Configuring Whole Server Migration for an Enterprise Deployment
  After you have prepared your domain for whole server migration or automatic service migration, you can configure Whole Server Migration for specific Managed Servers within a cluster.
- Configuring Automatic Service Migration in an Enterprise Deployment
  You may need to configure automatic service migration for specific services in an enterprise deployment.

About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment

Oracle WebLogic Server provides a migration framework that is an integral part of any highly available environment. The following sections provide more information about how this framework can be used effectively in an enterprise deployment.

- Understanding the Difference between Whole Server and Service Migration
- Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment
- Understanding Which Products and Components Require Whole Server Migration and Service Migration

Understanding the Difference between Whole Server and Service Migration

The Oracle WebLogic Server migration framework supports two distinct types of automatic migration:

- **Whole Server Migration**, where the Managed Server instance is migrated to a different physical system upon failure.

Whole server migration provides for the automatic restart of a server instance, with all its services, on a different physical machine. When a failure occurs in a server that is part of a cluster which is configured with server migration, the server is restarted on any of the other machines that host members of the cluster.

For this to happen, the servers must use a floating IP as listen address and the required resources (transactions logs and JMS persistent stores) must be available on the candidate machines.

See Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.

- **Service Migration**, where specific services are moved to a different Managed Server within the cluster.

  To understand service migration, it's important to understand *pinned services*.

  In a WebLogic Server cluster, most subsystem services are hosted homogeneously on all server instances in the cluster, enabling transparent failover from one server to another. In contrast, pinned services, such as JMS-related services, the JTA Transaction Recovery Service, and user-defined singleton services, are hosted on individual server instances within a cluster—for these services, the WebLogic Server migration framework supports failure recovery with service migration, as opposed to failover.

  See Understanding the Service Migration Framework in *Administering Clusters for Oracle WebLogic Server*.

Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment

Using Whole Server Migration (WSM) or Automatic Service Migration (ASM) in an Enterprise Deployment has implications in the infrastructure and configuration requirements.

The implications are:

- The resources used by servers must be accessible to both the original and failover system

  In its initial status, resources are accessed by the original server or service. When a server or service is failed over/restarted in another system, the same resources (such as external resources, databases, and stores) must be available in the failover system. Otherwise, the service cannot resume the same operations. It is for this reason, that both whole server and service migration require that all members of a WebLogic cluster have access to the same transaction and JMS persistent stores (whether the persistent store is file-based or database-based).

  Oracle allows you to use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster. Alternatively, you can use shared storage. When you configure persistent stores properly in the database or in shared storage, you must ensure that if a failover occurs (whole server migration or service migration), the failover system is able to access the same stores without any manual intervention.

- Leasing Datasource

  Both server migration and service migration (whether in static or dynamic clusters) require the configuration of a leasing datasource that is used by servers to store *alive* timestamps. These timestamps are used to determine the health of a server or service, and are key to the correct behavior of server and service migration (they are used to marks servers or services as *failed* and trigger failover).

> **Note:**
>
> Oracle does not recommend that you use consensus leasing for HA purposes.

- Virtual IP address

  In addition to shared storage, Whole Server Migration requires the procurement and assignment of a virtual IP address (VIP) for each individual server and the corresponding Virtual Host Name which is mapped to this IP and used as the listen address for the involved server. When a Managed Server fails over to another machine, the VIP is enabled in the failover node by Node Manager. Service migration does not require a VIP.

Since server migration requires a full restart of a managed server, it involves a higher failover latency than service migration. Table 20-1 summarizes the different aspects.

Table 20-1 Different Aspects of WSM and ASM

Cluster Protection	Failover Time	Capacity Planning	Reliability	Shared Storage/DB	VIP per Managed Server
WSM	4–5 mins	Full Server running	DB Leasing	Yes	Yes
ASM	30 secs	Mem/CPU of services	DB Leasing	Yes	No

Understanding Which Products and Components Require Whole Server Migration and Service Migration

Note that the table lists the recommended best practice. It does not preclude you from using Whole Server or Automatic Server Migration for those components that support it.

Component	Whole Server Migration (WSM)	Automatic Service Migration (ASM)
Oracle Identity Manager	YES	YES (Recommended)
Oracle Web Services Manager (OWSM)	NO	NO
Oracle SOA Suite	YES	YES (Recommended)

Creating a GridLink Data Source for Leasing

Whole Server Migration and Automatic Service Migration require a data source for the leasing table, which is a tablespace created automatically as part of the Oracle WebLogic Server schemas by the Repository Creation Utility (RCU).

> **Note:**
>
> To accomplish data source consolidation and connection usage reduction, you can reuse the `WLSSchemaDatasource` as is for database leasing. This datasource is already configured with the `FMW1221_WLS_RUNTIME` schema, where the leasing table is stored.

For an enterprise deployment, you should create a GridLink data source:

1. Log in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Data Sources**.
4. On the Summary of Data Sources page, click **New** and select **GridLink Data Source**, and enter the following:
   - Enter a logical name for the data source in the **Name** field. For example, **Leasing**.
   - Enter a name for **JNDI**. For example, **jdbc/leasing**.
   - For the Database Driver, select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
   - Click **Next**.
5. In the Transaction Options page, clear the **Supports Global Transactions** check box, and then click **Next**.
6. In the GridLink Data Source Connection Properties Options screen, select **Enter individual listener information** and click **Next**.
7. Enter the following connection properties:
   - **Service Name**: Enter the service name of the database with lowercase characters. For a GridLink data source, you must enter the Oracle RAC service name. For example:

     ```
     iamedg.example.com
     ```

   - **Host Name and Port**: Enter the SCAN address and port for the RAC database, separated by a colon. For example:

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

     Click **Add** to add the host name and port to the list box below the field.

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

     ```
     SQL>show parameter remote_listener;

     NAME                     TYPE         VALUE
     ```

```
-------------------------------------------------

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

> **Note:**
>
> For Oracle Database 11*g* Release 1 (11.1), use the virtual IP and port of each database instance listener, for example:
>
> ```
> dbhost1-vip.mycompany.com (port 1521)
> ```
>
> and
>
> ```
> dbhost2-vip.mycompany.com (1521)
> ```
>
> For Oracle Database 10*g*, use multi data sources to connect to an Oracle RAC database.

- **Database User Name**: Enter the following:

  ```
  FMW1221_WLS_RUNTIME
  ```

  In this example, FMW1221 is the prefix you used when you created the schemas as you prepared to configure the initial enterprise manager domain.

  Note that in previous versions of Oracle Fusion Middleware, you had to manually create a user and tablespace for the migration leasing table. In Fusion Middleware 12*c* (12.2.1), the leasing table is created automatically when you create the WLS schemas with the Repository Creation Utility (RCU).

- **Password**: Enter the password you used when you created the WLS schema in RCU.
- **Confirm Password**: Enter the password again and click **Next**.

8. On the Test GridLink Database Connection page, review the connection parameters and click **Test All Listeners**.

   Here is an example of a successful connection notification:

   ```
   Connection test for
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=db-
   scan.example.com)
   (PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=iamedg.example.com))) succeeded.
   ```

   Click **Next**.

9. In the ONS Client Configuration page, do the following:

   - Select **FAN Enabled** to subscribe to and process Oracle FAN events.
   - Enter the SCAN address in the **ONS Host and Port** field, and then click **Add**.

     This value should be the ONS host and ONS remote port for the RAC database. To find the ONS remote port for the database, you can use the following command on the database host:

     ```
     [orcl@db-scan1 ~]$ srvctl config nodeapps -s

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

   - Click **Next**.

> **Note:**
>
> For Oracle Database 11*g* Release 1 (11.1), use the hostname and port of each database's ONS service, for example:
>
> ```
> custdbhost1.example.com (port 6200)
> ```
>
> and
>
> ```
> custdbhost2.example.com (6200)
> ```

10. On the Test ONS Client Configuration page, review the connection parameters and click **Test All ONS Nodes**.

    Here is an example of a successful connection notification:

    ```
    Connection test for db-scan.example.com:6200 succeeded.
    ```

    Click **Next**.
11. In the Select Targets page, select the cluster that you are configuring for Whole Server Migration or Automatic Service Migration, and then select **All Servers in the cluster**.
12. Click **Finish**.
13. Click **Activate Changes**.

Configuring Whole Server Migration for an Enterprise Deployment

After you have prepared your domain for whole server migration or automatic service migration, you can configure Whole Server Migration for specific Managed Servers within a cluster.

> **Note:**
>
> As mentioned earlier, for migration to work, servers must use a virtual hostname that matches a floating IP, as the listen address. You can specify the listen address directly in the Configuration Wizard or update it in the administration console.

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

Editing the Node Manager's Properties File to Enable Whole Server Migration

Use the section to edit the Node Manager properties file on the two nodes where the servers are running.

1. Locate and open the following file with a text editor:

   ```
   MSERVER_HOME/nodemanager/nodmeanager.properties
   ```

2. If not done already, set the `StartScriptEnabled` property in the `nodemanager.properties` file to true.

   This is required to enable Node Manager to start the managed servers.

3. Add the following properties to the `nodemanager.properties` file to enable server migration to work properly:

   - `Interface`

     ```
     Interface=eth0
     ```

     This property specifies the interface name for the floating IP (`eth0`, for example).

     > **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, the valid values in Linux environments are `eth0`, `eth1`, or, `eth2`, `eth3`, `eth`*`n`*, depending on the number of interfaces configured.

   - `NetMask`

     ```
     NetMask=255.255.255.0
     ```

     This property specifies the net mask for the interface for the floating IP.

   - `UseMACBroadcast`

     ```
     UseMACBroadcast=true
     ```

     This property specifies whether to use a node's MAC address when sending ARP packets, that is, whether to use the `-b` flag in the arping command.

4. Restart the Node Manager.

5. Verify in the output of Node Manager (the shell where the Node Manager is started) that these properties are in use. Otherwise, problems may occur during migration. The output should be similar to the following:

   ```
   ...
   SecureListener=true
   LogCount=1
   eth0=*,NetMask=255.255.255.0
   ...
   ```

Setting Environment and Superuser Privileges for the wlsifconfig.sh Script

Use this section to set the environment and superuser privileges for the `wlsifconfig.sh` script, which is used to transfer IP addresses from one machine to another during migration. It must be able to run `ifconfig`, which is generally only available to superusers.

For more information about the `wlsifconfig.sh` script, see Configuring Automatic Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.

Refer to the following sections for instructions on preparing your system to run the `wlsifconfig.sh` script.

- Setting the PATH Environment Variable for the wlsifconfig.sh Script
- Granting Privileges to the wlsifconfig.sh Script

Setting the PATH Environment Variable for the wlsifconfig.sh Script

Ensure that the commands listed in the following table are included in the PATH environment variable for each host computers.

File	Directory Location
`wlsifconfig.sh`	*MSERVER_HOME*/bin/server_migration
`wlscontrol.sh`	*WL_HOME*/common/bin
`nodemanager.domains`	*MSERVER_HOME*/nodemanager

Granting Privileges to the wlsifconfig.sh Script

Grant *sudo* privilege to the operating system user (for example, `oracle`) with no password restriction, and grant execute privilege on the `/sbin/ifconfig` and `/sbin/arping` binaries.

> **Note:**
>
> For security reasons, `sudo` should be restricted to the subset of commands required to run the `wlsifconfig.sh` script.
>
> Ask the system administrator for the *sudo* and system rights as appropriate to perform this required configuration task.

The following is an example of an entry inside */etc/sudoers* granting *sudo* execution privilege for `oracle` to run `ifconfig` and `arping`:

```
Defaults:oracle !requiretty
oracle ALL=NOPASSWD: /sbin/ifconfig,/sbin/arping
```

Configuring Server Migration Targets

To configure migration in a cluster:

1. Sign in to the Oracle WebLogic Server Administration Console.

2. In the Domain Structure window, expand **Environment** and select **Clusters**. The Summary of Clusters page is displayed.
3. Click the cluster for which you want to configure migration in the Name column of the table.
4. Click the **Migration** tab.
5. Click **Lock & Edit**.
6. Select **Database** as Migration Basis. From the drop-down list, select **Leasing** as Data Source For Automatic Migration.
7. Under **Candidate Machines For Migratable Server**, in the Available filed, select the Managed Servers in the cluster and click the right arrow to move them to **Chosen**.
8. Click **Save**.
9. Set the Candidate Machines for Server Migration. You must perform this task for all of the managed servers as follows:
   a. In Domain Structure window of the Oracle WebLogic Server Administration Console, expand **Environment** and select **Servers**.
   b. Select the server for which you want to configure migration.
   c. Click the **Migration** tab.
   d. Select **Automatic Server Migration Enabled** and click **Save**.

      This enables the Node Manager to start a failed server on the target node automatically.
   e. In the **Available** field, located in the Migration Configuration section, select the machines to which to allow migration and click the right arrow.

      In this step, you are identifying the host to which the Managed Server should failover if the current host is unavailable. For example, for the Managed Server on the HOST1, select HOST2; for the Managed Server on HOST2, select HOST1.

   > **Tip:**
   >
   > Click **Customize this table** in the Summary of Servers page, move Current Machine from the Available Window to the Chosen window to view the machine on which the server is running. This is different from the configuration if the server is migrated automatically.

10. Click **Activate Changes**.
11. Restart the Administration Server and the servers for which server migration has been configured.

Testing Whole Server Migration

Perform the steps in this section to verify that automatic whole server migration is working properly.

To test from Node 1:

1. Stop the managed server process.

   ```
   kill -9 pid
   ```

pid specifies the process ID of the managed server. You can identify the *pid* in the node by running this command:

2. Watch the Node Manager console (the terminal window where you performed the kill command): you should see a message indicating that the managed server's floating IP has been disabled.

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

4. After node manager restarts the server and before it reaches *Running* state, kill the associated process again.

   Node Manager should log a message indicating that the server will not be restarted again locally.

   > **Note:**
   >
   > The number of restarts required is determined by the `RestartMax` parameter in the following configuration file:
   >
   > The default value is `RestartMax=2`.

To test from Node 2:

1. Watch the local Node Manager console. After 30 seconds since the last try to restart the managed server on Node 1, Node Manager on Node 2 should prompt that the floating IP for the managed server is being brought up and that the server is being restarted in this node.

2. Access a product URL by using the same IP address. If the URL is successful, then the migration was successful.

Verification From the Administration Console

You can also verify migration using the Oracle WebLogic Server Administration Console:

1. Log in to the Administration Console.
2. Click **Domain** on the left console.
3. Click the **Monitoring** tab and then the **Migration** subtab.

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

> **Note:**
>
> After a server is migrated, to fail it back to its original machine, stop the managed server from the Oracle WebLogic Administration Console and then start it again. The appropriate Node Manager starts the managed server on the machine to which it was originally assigned.

Configuring Automatic Service Migration in an Enterprise Deployment

You may need to configure automatic service migration for specific services in an enterprise deployment.

- Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster
- Configuring Automatic Service Migration for Static Clusters
- Configuring Automatic Service Migration for Dynamic Clusters

Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster

Before you can configure automatic service migration, you must verify the leasing mechanism and data source that will be used by the automatic service migration feature:

> **Note:**
>
> The following procedure assumes you have already created the Leasing data source, as described in Creating a GridLink Data Source for Leasing.

1. Log in to the Oracle WebLogic Server Administration Console.
2. Click **Lock & Edit**.
3. In the Domain Structure window, expand **Environment** and select **Clusters**.

   The Summary of Clusters page appears.
4. In the **Name** column of the table, click the cluster for which you want to configure migration.
5. Click the **Migration** tab.
6. Verify that **Database** is selected in the **Migration Basis** drop-down menu.
7. From the **Data Source for Automatic Migration** drop-down menu, select the Leasing data source that you created in Creating a GridLink Data Source for Leasing.
8. Click **Save**.
9. Activate changes.
10. Restart the managed servers for the changes to be effective. If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

Configuring Automatic Service Migration for Static Clusters

After you have configured the leasing for the cluster as described in Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster, you can configure automatic service migration for specific services in an enterprise deployment. The following sections explain how to configure and validate Automatic Service Migration for static clusters.

- Changing the Migration Settings for the Managed Servers in the Cluster
- About Selecting a Service Migration Policy
- Setting the Service Migration Policy for Each Managed Server in the Cluster
- Validating Automatic Service Migration in Static Clusters
- Failing Back Services After Automatic Service Migration

Changing the Migration Settings for the Managed Servers in the Cluster

After you set the leasing mechanism and data source for the cluster, you can then enable automatic JTA migration for the Managed Servers that you want to configure for service migration. Note that this topic applies only if you are deploying JTA services as part of your enterprise deployment.

To change the migration settings for the Managed Servers in each cluster:

1. If you haven’t already, log in to the Administration Console, and click **Lock & Edit**.
2. In the Domain Structure pane, expand the **Environment** node and then click **Servers**.

   The Summary of Servers page appears.
3. Click the name of the server you want to modify in **Name** column of the table.

   The settings page for the selected server appears and defaults to the Configuration tab.
4. Click the **Migration** tab.
5. From the **JTA Migration Policy** drop-down menu, select **Failure Recovery**.
6. In the **JTA Candidate Servers** section of the page, select the Managed Servers in the **Available** list box, and then click the move button to move them into the **Chosen** list box.
7. In the **JMS Service Candidate Servers** section of the page, select the Managed Servers in the **Available** list box, and then click the move button to move them into the **Chosen** list box.
8. Click **Save**.
9. Activate the changes.
10. Restart the managed servers and the Administration Server for the changes to be effective. If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

About Selecting a Service Migration Policy

When you configure Automatic Service Migration, you select a Service Migration Policy for each cluster. This topic provides guidelines and considerations when selecting the Service Migration Policy.

For example, products or components running singletons or using Path services can benefit from the **Auto-Migrate Exactly-Once** policy. With this policy, if at least one Managed Server in the candidate server list is running, the services hosted by this migratable target are active somewhere in the cluster if servers fail or are administratively shut down (either gracefully or forcibly). This can cause multiple homogenous services to end up in one server on startup.

When you use this policy, you should monitor the cluster startup to identify what servers are running on each server. You can then perform a manual failback, if necessary, to place the system in a balanced configuration.

Other Fusion Middleware components are better suited for the **Auto-Migrate Failure-Recovery Services** policy.

See Policies for Manual and Automatic Service Migration in *Administering Clusters for Oracle WebLogic Server*.

Setting the Service Migration Policy for Each Managed Server in the Cluster

After you modify the migration settings for each server in the cluster, you can then identify the services and set the migration policy for each Managed Server in the cluster, using the WebLogic Administration Console:

1. If you have not already, log in to the Administration Console, and click **Lock & Edit**.
2. In the Domain Structure pane, expand **Environment**, then expand **Clusters**, then select **Migratable Targets**.
3. Click the name of the first Managed Server in the cluster.
4. Click the **Migration** tab.
5. From the **Service Migration Policy** drop-down menu, select the appropriate policy for the cluster.

   See About Selecting a Service Migration Policy.
6. Click **Save**.
7. Repeat steps 2 through for each of the additional Managed Servers in the cluster.
8. Activate the changes.
9. Restart the managed servers for the changes to be effective. If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

Validating Automatic Service Migration in Static Clusters

After you configure automatic service migration for your cluster and Managed Servers, validate the configuration, as follows:

1. If you have not already done so, log in to the Administration Console.
2. In the Domain Structure pane, expand **Environment**, and then expand **Clusters**.
3. Click **Migratable Targets**.
4. Click the **Control** tab.

   The console displays a list of migratable targets and their current hosting server.
5. In the Migratable Targets table, select a row for the one of the migratable targets.
6. Note the value in the **Current Hosting Server** column.
7. Use the operating system command line to stop the first Managed Server.

   Use the following command to end the Managed Server Process and simulate a crash scenario:

   ```
   kill -9 pid
   ```

   In this example, replace *pid* with the process ID (PID) of the Managed Server. You can identify the PID by running the following UNIX command:

   ```
   ps -ef | grep managed_server_name
   ```

> **Note:**
>
> After you kill the process, the Managed Server might be configured to start automatically. In this case, you must kill the second process using the `kill -9` command again.

8. Watch the terminal window (or console) where the Node Manager is running.

   You should see a message indicating that the selected Managed Server has failed. The message is similar to the following:

   ```
   <INFO> <domain_name> <server_name>
   <The server 'server_name' with process id 4668 is no longer alive; waiting for the
   process to die.>
   <INFO> <domain_name> <server_name>
   <Server failed during startup. It may be retried according to the auto restart
   configuration.>
   <INFO> <domain_name> <server_name>
   <Server failed but will not be restarted because the maximum number of restart
   attempts has been exceeded.>
   ```

9. Return to the Oracle WebLogic Server Administration Console and refresh the table of migratable targets; verify that the migratable targets are transferred to the remaining, running Managed Server in the cluster:

   - Verify that the Current Hosting Server for the process you killed is now updated to show that it has been migrated to a different host.
   - Verify that the value in the **Status of Last Migration** column for the process is *Succeeded*.

10. Open and review the log files for the Managed Servers that are now hosting the services; look for any JTA or JMS errors.

> **Note:**
>
> For JMS tests, it is a good practice to get message counts from destinations and make sure that there are no stuck messages in any of the migratable targets:
>
> For example, for uniform distributed destinations (UDDs):
>
> a. Access the JMS Subdeployment module in the Administration Console:
>
>   In the Domain Structure pane, select **Services**, then **Messaging**, and then **JMS Modules**.
>
> b. Click the JMS Module.
>
> c. In the Summary of Resources table, click **Destinations**, and then click the **Monitoring** tab.
>
> d. Review the **Messages Total** and **Messages Pending** values. Click **Customize table** to add these columns to the table, if these values do not appear in the table.

Failing Back Services After Automatic Service Migration

When Automatic Service Migration occurs, Oracle WebLogic Server does not support failing back services to their original server when a server is back online and rejoins the cluster.

As a result, after the Automatic Service Migration migrates specific JMS services to a backup server during a fail-over, it does not migrate the services back to the original server after the original server is back online. Instead, you must migrate the services back to the original server manually.

To fail back a service to its original server, follow these steps:

1. If you have not already done so, in the Change Center of the Administration Console, click **Lock & Edit.**
2. In the Domain Structure tree, expand **Environment**, expand **Clusters**, and then select **Migratable Targets**.
3. To migrate one or more migratable targets at once, on the Summary of Migratable Targets page:
   a. Click the **Control** tab.
   b. Use the check boxes to select one or more migratable targets to migrate.
   c. Click **Migrate**.
   d. Use the **New hosting server** drop-down to select the original Managed Server.
   e. Click **OK**.

      A request is submitted to migrate the JMS-related service. In the Migratable Targets table, the Status of Last Migration column indicates whether the requested migration has succeeded or failed.
   f. Release the edit lock after the migration is successful.

Configuring Automatic Service Migration for Dynamic Clusters

After you have configured the leasing for the cluster as described in Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster, you can continue with the Service Migration configuration.

Dynamic Clusters simplify the configuration for service migration because the services are targeted to the entire cluster. However, you still have to configure the migration policy at the custom persistent store level and for the JTA service. These policies determine the migration behavior of JMS and JTA services, respectively.

- About Selecting a Service Migration Policy for Dynamic Clusters
- Changing the Migration Settings for the Persistent Stores
- Changing the Migration Settings for the JTA Service
- Validating Automatic Service Migration in Dynamic Clusters
- Failing Back Services After Automatic Service Migration

About Selecting a Service Migration Policy for Dynamic Clusters

When you configure service migration for dynamic clusters, you select a Service Migration Policy for each persistent store. This topic provides guidelines and considerations when you select the Service Migration Policy. The following options are available:

- **Off**: Disables migration and restart support for cluster-targeted JMS service objects, including the ability to restart a failed persistent store instance and its associated services. You cannot combine this policy with the Singleton Migration Policy.

- **On-Failure**: Enables automatic migration and restart of instances on the failure of a subsystem Service or the WebLogic Server instance, including automatic fail-back and load balancing of instances.
- **Always**: Provides the same behavior as On-Failure and automatically migrates instances even if a graceful shutdown or a partial cluster start occurs.

Products or components that run singletons or use Path services can benefit from the **Always** policy. With this policy, if at least one Managed Server is running, the instances remain active somewhere in the cluster if servers fail or are administratively shut down (either gracefully or forcibly). This type of failure or shutdown can cause multiple homogenous services to end up in one server on startup.

Other Fusion Middleware components are better suited for the **On-Failure** policy.

Based on these guidelines, the following policies are recommended for an Oracle SOA Suite enterprise topology:

- SOA_Cluster: On-Failure
- OSB_Cluster: On-Failure
- MFT_Cluster: On-Failure

For information about the JMS configuration for high availability, see Simplified JMS Cluster and High Availability Configuration.

Changing the Migration Settings for the Persistent Stores

After you choose the migration policy for each cluster, you can identify the persistent stores of the cluster and set the migration policy for each cluster by using the WebLogic Administration Console:

1. Log in to the Administration Console, if you have not already done so, and click **Lock & Edit**.
2. In the Domain Structure pane, expand **Environment**, expand **Services**, and then select **Persistent Stores**.
3. Click the name of the **Persistent Store** that you want to modify.

   > **Note:**
   >
   > When you use JDBC persistent stores, additional unused File Stores are automatically created but are not targeted to your clusters. Ignore these File Stores.

4. Click the **High Availability** tab.
5. From the **Migration Policy** drop-down menu, select the appropriate policy for the cluster. See About Selecting a Service Migration Policy for Dynamic Clusters.
6. Click **Save**.
7. Repeat steps 2 through 6 for each additional persistent store in the cluster.
8. Click **Activate Changes**.
9. Restart the managed servers for the changes to be effective. If you are configuring other aspects of service migration in the same configuration change session, you can use a final unique restart to reduce downtime.

Changing the Migration Settings for the JTA Service

You must set the appropriate migration policy for the JTA service in each server so that any member in the cluster can resume the XA logs in the event of a failure or shutdown of one of the members of the dynamic cluster. To set the migration policy for the servers in a dynamic cluster, follow these steps:

1. Log in to the FMW Control Console by accessing `ADMINVHN:7001/console`and by using the required credentials.
2. Click the lock icon on the upper right corner and click **Lock & Edit**.
3. On the target navigation tree on the left, select the relevant domain.
4. Click **Weblogic Domain** > **Environment** > **Server templates**.
5. Click the relevant template and then, click the **Migration** tab.
6. From the **JTA Migration Policy** drop-down list, select the required migration policy for the service. The settings required for each SOA component is as follows. (Some may not be shown, depending on what has been installed.):
   - SOA_Cluster: Failure Recovery
   - OIM_Cluster: Failure Recovery
7. Click **Save**.
8. Click the lock icon on the upper right corner and click **Activate Changes**.
9. Restart the managed servers and the Administration Server for the changes to be effective.

Validating Automatic Service Migration in Dynamic Clusters

After you configure service migration for your dynamic cluster, validate the configuration, as follows:

1. Log in to the Administration Console, if you have not already done so.
2. In the Domain Structure pane, select **Environment**, and then **Clusters**.
3. Click in the cluster where you want to verify the service migration.
4. Click the **Monitoring** tab, then **Health**.

   The console displays a list of the servers of the cluster and their state.
5. Expand each managed server and verify that its persistent stores are okay.
6. In Domain Structure pane, select **Services** > **Messaging** > **JMS Servers**.
7. Click on one of the JMS Servers of the cluster, and then click the **Monitoring** tab.

   Verify that you see two instances (one per dynamic server) and each instance is running on one of the dynamic servers.
8. Use the operating system command line to stop the first Managed Server. Use the following command to end the Managed Server process and simulate a crash scenario:

   ```
   kill -9 pid
   ```

In this example, replace *pid* with the process ID (PID) of the Managed Server. You can identify the *PID* by running the following UNIX command:

```
ps -ef | grep managed_server_name
```

> **Note:**
>
> You can configure the Managed Server to start automatically after you initially kill the process. In this case, you must kill the second process by using the `kill -9` command again.

9. Watch the terminal window (or console) where the Node Manager is running.

   You see a message indicating that the selected Managed Server has failed. The message appears as follows:

   ```
   <INFO> <domain_name> <server_name>
   <The server 'server_name' with process id 4668 is no longer alive; waiting for the
   process to die.>
   <INFO> <domain_name> <server_name>
   <Server failed during startup. It may be retried according to the auto restart
   configuration.>
   <INFO> <domain_name> <server_name>
   <Server failed but will not be restarted because the maximum number of restart
   attempts has been exceeded.>
   ```

10. Return to the Oracle WebLogic Server Administration Console and refresh the table of **Cluster** > **Monitoring** > **Health**. Verify that the persistent stores are now running in the remaining Managed Server that is still running.

11. In Domain Structure pane, select **Services** > **Messaging** > **JMS Servers**.

12. Click on one of the JMS Servers of the cluster, and then click the **Monitoring** tab.

    Verify that both the instances continue to run on the remaining Managed Server that is still running.

13. Open and review the log files for the Managed Servers that are now hosting the services. Look for any JTA or JMS errors.

> **Note:**
>
> For JMS tests, it is a good practice to get message counts from destinations and ensure that messages are not stuck in the migratable targets. For example, for uniform distributed destinations (UDDs):
>
> a. Access the JMS Subdeployment module in the Administration Console.
>
> b. In the Domain Structure pane, select **Services** > **Messaging** > **JMS Modules**.
>
> c. Click the JMS Module.
>
> d. In the Summary of Resources table, click **Destinations**, and then click the **Monitoring** tab. Review the **Messages Total** and **Messages Pending** values.
>
> Click **Customize table** to add these columns to the table, if these values do not appear in the table.

14. Review the logs. The messages appear as follows in the remaining server:

```
<Info> <Cluster> <soahost1> <WLS_SOA1> <[STANDBY] ExecuteThread:
'43' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>>
<> <49c99f17-a5d6-487d-a710-65eef0262ebc-0000063c> <1489481002608>
<[severity-value: 64] [rid: 0] [partition-id: 0] [partition-name: DOMAIN]
> <BEA-000189>
<The Singleton Service UMSJMSJDBCStore_auto_1_WLS_SOA2 is now active on
this server.>

<Info> <Cluster> <soahost1> <WLS_SOA1> <[STANDBY] ExecuteThread:
'43' for queue: 'weblogic.kernel.Default (self-tuning)'> <<WLS Kernel>>
<> <49c99f17-a5d6-487d-a710-65eef0262ebc-0000063c> <1489481002609>
<[severity-value: 64] [rid: 0] [partition-id: 0] [partition-name: DOMAIN]
> <BEA-003130>
<UMSJMSJDBCStore_auto_1_WLS_SOA2 successfully activated on server
WLS_SOA1.>
```

For more information, you can debug with the following flags:

```
-Dweblogic.debug.DebugSingletonServices=true -
Dweblogic.debug.DebugServerMigration=true
```

Failing Back Services After Automatic Service Migration

With dynamic clustering, when a distributed instance is migrated from its preferred server, it tries to fail back when the preferred server is restarted. Therefore, after the service migration process migrates specific persistent store services to a backup server during a failover, it migrates the services back to the original server after the original server is back online.

21

Scaling Procedures for an Enterprise Deployment

The scaling procedures for an enterprise deployment include scale out, scale in, scale up, and scale down. During a scale-out operation, you add managed servers to new nodes. You can remove these managed servers by performing a scale in operation. During a scale-up operation, you add managed servers to existing hosts. You can remove these servers by performing a scale-down operation.

This chapter describes the procedures to scale out/in and scale up/down static and dynamic clusters.

- Scaling Out the Topology
  When you scale out the topology, you add new managed servers to new nodes.
- Scaling Up the Topology
  When you scale up the topology, you add new managed servers to the existing hosts.
- OAM Specific Scaling Actions
  This section briefs about registering any new managed servers with access manager and Webgate Agents.

Scaling Out the Topology

When you scale out the topology, you add new managed servers to new nodes.

This section describes the procedures to scale out the Identity Management topology with static and dynamic clusters.

> **Note:**
>
> The dynamic clusters are applicable only for the Governance Domain components like, Oracle SOA Suite and Oracle Identity Governance.

- Scaling Out the Topology for Static Clusters
- Scaling Out the Topology for Dynamic Clusters

Scaling Out the Topology for Static Clusters

This section lists the prerequisites, explains the procedure to scale out the topology with static clusters, describes the steps to verify the scale-out process, and finally the steps to scale down (shrink).

- Prerequisites for Scaling Out
- Scaling Out a Static Cluster
- Verifying the Scale Out of Static Clusters
- Scaling in the Topology for Static Clusters

Prerequisites for Scaling Out

Before you perform a scale out of the topology, you must ensure that you meet the following requirements:

- The starting point is a cluster with managed servers already running.
- The new node can access the existing home directories for WebLogic Server and Governance. Use the existing installations in shared storage. You do not need to install WebLogic Server or IDM binaries in a new location. However, you do need to run `pack` and `unpack` commands to bootstrap the domain configuration in the new node.
- It is assumed that the cluster syntax is used for all internal RMI invocations, JMS adapter, and so on.

Scaling Out a Static Cluster

The steps provided in this procedure use the IDM EDG topology as a reference. Initially there are two application tier hosts (OAMHOST1 and OAMHOST2, or OIMHOST1 and OIMHOST2), each running one managed server of each cluster. A new host HOST3 is added to scale up the clusters with a third managed server. `WLS_XYZ`*n* is the generic name given to the new managed server that you add to the cluster. Depending on the cluster that is being extended and the number of existing nodes, the actual names will be `WLS_OAM3`, `WLS_AMA3`, `WLS_SOA3`, and so on.
To scale out the cluster, complete the following steps:

1. On the new node, mount the existing FMW Home, which should include the IDM installation and the domain directory. Ensure that the new node has access to this directory, similar to the rest of the nodes in the domain. Also, ensure that the FMW Home you mount is the one associated with the domain your are extending.
2. Locate the inventory in the shared directory (for example, `/u01/oracle/products/oraInventory`), per Oracle's recommendation. So you do not need to attach any home, but you may want to execute the script: `/u01/oracle/products/oraInventory/createCentralInventory.sh`.

   This command creates and updates the local file `/etc/oraInst.loc` in the new node to point it to the oraInventory location.

   If there are other inventory locations in the new host, you can use them, but `/etc/oraInst.loc` file must be updated accordingly for updates in each case.
3. Update the `/etc/hosts` files to add the name of the new host (unless you are using DNS), as described in Verifying IP Addresses and Host Names in DNS or Hosts File. If you are using host aliases such as OIMHOST, then ensure that you add an entry for that host.

   For example:

   ```
   10.229.188.204 host1-vip.example.com host1-vip ADMINVHN
   10.229.188.205 host1.example.com host1 OIMHOST1
   10.229.188.206 host2.example.com host2 OIMHOST2
   10.229.188.207 host3.example.com host3 WEBHOST1
   10.229.188.208 host4.example.com host4 WEBHOST2
   10.229.188.209 host5.example.com host5 OIMHOST3
   ```
4. Log in to the Oracle WebLogic Administration Console to create a new machine:
   a. Go to **Environment** > **Machines**.
   b. Click **New** to create a new machine for the new node.

   c. Set **Name** to OIMHOST*n* or OAMHOST*n*.
   d. Set **Machine OS** to Linux.
   e. Click **Next**.
   f. Set **Type** to Plain.
   g. Set **Listen Address** to the new host name. For example, OIMHOST3.
   h. Click **Finish**, and then click **Activate Changes**.

5. Use the Oracle WebLogic Server Administration Console to clone the first managed server in the cluster into a new managed server.
   a. In the Change Center section, click **Lock & Edit**.
   b. Go to **Environment** > **Servers**.
   c. Select the first managed server in the cluster to scale out and click **Clone**.
   d. Use Details of the Cluster to be Scaled Out to set the correspondent name, listen address, and listen port, depending on the cluster that you want to scale out.
   e. Click the new managed server, and then **Configuration** > **General**.
   f. Update the **Machine** from OIMHOST1 to OIMHOST*n*.
   g. Click **Save**, and then click **Activate Changes**.

Table 21-1 Details of the Cluster to be Scaled Out

Cluster to Scale Out	Server to Clone	New Server Name	Server Listen Address	Server Listen Port
WSM-PM_Cluster	WLS_WSM1	WLS_WSM*n*	OIMHOST*n*	7010
SOA_Cluster	WLS_SOA1	WLS_SOA*n*	OIMHOST*n*	8001
OIM_Cluster	WLS_OIM1	WLS_OIM*n*	OIMHOST*n*	14000
OAM_Cluster	WLS_OAM1	WLS_OAM*n*	OAMHOST*n*	14100
AMA_Cluster	WLS_AMA1	WLS_AMA*n*	OAMHOST*n*	14150

6. Update the deployment Staging Directory Name of the new server, as described in Modifying the Upload and Stage Directories to an Absolute Path.
7. Create a new key certificate and update the private key alias of the server, as described in Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer.
8. By default, the cloned server uses default store for TLOGs. If the rest of the servers in the cluster that you are scaling-out are using TLOGs in JDBC persistent store, update the TLOG persistent store of the new managed server:
   a. Go to **Environment** > **Servers** > **WLS_XYZn** > **Configuration** > **Services**.
   b. Change **Transaction Log Store** to JDBC.
   c. Change **Data Source** to *WLSSchemaDatasource*.
   d. Click **Save**, and then click **Activate Changes**.

   Use the following table to identify the clusters that use JDBC TLOGs by default:

Table 21-2   The Name of Clusters that Use JDBC TLOGs by Default

Cluster to Scale Out	New Server Name	TLOG Persistent Store
WSM-PM_Cluster	WLS_WSM*n*	Default (file)
SOA_Cluster	WLS_SOA*n*	JDBC
OIM_Cluster	WLS_OIM*n*	Default JDBC
OAM_Cluster	WLS_OAM*n*	Not Applicable
AMA_Cluster	WLS_AMA*n*	Not Applicable

9. If the cluster that you are scaling out is configured for automatic service migration, update the **JTA Migration Policy** to the required value.

   a. Go to **Environment** > **Servers** > **WLS_XYZn** > **Configuration** > **Migration**.

   b. Use Table 21-3 to set the recommended JTA Migration Policy depending on the cluster that you want to scale out.

   Table 21-3   The Recommended JTA Migration Policy for the Cluster to be Scaled Out

Cluster to Scale Out	New Server Name	JTA Migration Policy
WSM-PM_Cluster	WLS_WSM*n*	Manual
SOA_Cluster	WLS_SOA*n*	Failure Recovery
OAM_Cluster	WLS_OAM*n*	Not Applicable
AMA_Cluster	WLS_AMA*n*	Not Applicable
OIM_Cluster	WLS_OIM*n*	Failure Recovery

   c. Click **Save**, and then click **Activate Changes**.

   d. For the rest of the servers already existing in the cluster, update the list of **JTA candidate servers** for JTA migration to include the new server.

   - Go to **Environment** > **Servers** > **server** > **Configuration** > **Migration**.
   - Go to **JTA Candidate Servers**: leave the list empty (leaving it empty because all server in the cluster are JTA candidate servers).
   - Click **Save**, and then click **Activate Changes**. Although you need to restart the servers for this change to be effective, you can do a unique restart later, after you complete all the required configuration changes.

10. If the cluster you are scaling out is configured for automatic service migration, use the Oracle WebLogic Server Administration Console to update the automatically created WLS_XYZn (migratable) with the recommended migration policy, because by default it is set to **Manual Service Migration Only**.

    Use the following table for the list of migratable targets to update:

Table 21-4   The Recommended Migratable Targets to Update

Cluster to Scale Out	Migratable Target to Update	Migration Policy
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	WLS_SOA*n* (migratable)	Auto-Migrate Failure Recovery Services

Table 21-4 (Cont.) The Recommended Migratable Targets to Update

Cluster to Scale Out	Migratable Target to Update	Migration Policy
OIM_Cluster	WLS_OIM*n* (migratable)	Auto-Migrate Failure Recovery Services
OAM_Cluster	Not applicable	Not applicable
AMA_Cluster	Not applicable	Not applicable

a. Go to **Environment > Migratable Servers**.

b. Click **Lock & Edit**.

c. Click WLS_XYZ3 (migratable).

d. Go to the tab **Configuration** > **Migration**.

e. Change the **Service Migration Policy** to the value listed in the table.

f. Leave the **Constrained Candidate Server** list blank in case there are chosen servers. If no servers are selected, you can migrate this migratable target to any server in the cluster.

g. Click **Save**, and then click **Activate Changes**.

11. Update the **Constrained Candidate Server** list in the existing migratable servers in the cluster that you are scaling because by default they are pre-populated with only WLS_XYZ1 and WLS_XYZ2 servers.

    a. Go to each migratable server.

    b. Go to the tab **Configuration** > **Migration** > **Constrained Candidate Server**.

       You can leave the server list blank to make these migratable targets migrate to any server in this cluster, including the newly created managed server.

       Use the following table to identify the migratable servers that have to be updated:

Table 21-5 The Existing Migratable Targets to Update

Cluster to Scale Out	Existing Migratable Target to Update	Constrained Candidate Server
WSM-PM_Cluster	Not applicable	Leave empty
SOA_Cluster	WLS_SOA1 (migratable) WLS_SOA2 (migratable)	Leave empty
OIM_Cluster	WLS_OIM1 (migratable) WLS_OIM2 (migratable)	Leave empty

    c. Click **Save**, and then click **Activate Changes**. Although you need to restart the servers for this change to be effective, you can do a unique restart later, after you complete all the required configuration changes.

12. Create the required persistent stores for the JMS servers.

    a. Log in to WebLogic Console and go to **Services** > **Persistent Stores**.

    b. Click **New** and select **Create JDBCStore**.

    Use the following table to create the required persistent stores:

> **Note:**
>
> The number in names and prefixes in the existing resources were assigned automatically by the Configuration Wizard during the domain creation. For example:
>
> - BPMJMSJDBCStore_auto_1 — soa_1
> - BPMJMSJDBCStore_auto_2 — soa_2
> - JDBCStore-OIM_auto_1 - oim1
> - JDBCStore-OIM_auto_2 - oim2
> - SOAJMSJDBCStore_auto_1 - soa_1
> - SOAJMSJDBCStore_auto_2 - soa_2
> - UMSJMSJDBCStore_auto_1 - soa_1
> - UMSJMSJDBCStore_auto_2 - soa_2
>
> So review the existing prefixes and select a new and unique prefix and name for each new persistent store.
>
> To avoid naming conflicts and simplify the configuration, new resources are qualified with the *scaled* tag and are shown here as an example.

Table 21-6 The New Resources Qualified with the Scaled Tag

Cluster to Scale Out	Persistent Store	Prefix Name	Data Source	Target
WSM-PM_Cluster	Not applicable	Not applicable	Not applicable	Not applicable
SOA_Cluster	UMSJMSJDBCStore_soa_scaled_3	soaums_scaled_3	WLSSchemaData Sourc	WLS_SOA3 (migratable)
	SOAJMSJDBCStore_ soa_scaled_3	soajms_scaled_3	WLSSchemaData Sourc	WLS_SOA3 (migratable)
	BPMJMSJDBCStore_ soa_scaled_3	soabpm_scaled_3	WLSSchemaData Source	WLS_SOA3 (migratable)
	(only when you use Insight) ProcMonJMSJDBCStore_soa_scaled_3	soaprocmon_scaled_3	WLSSchemaData Source	WLS_SOA3 (migratable)
OIM_Cluster	NA	JDBCStore-OIM_scaled_3	WLSSchemaData Source	WLS_OIM3 (migratable)

13. Create the required JMS Servers for the new managed server.

    a. Go to **WebLogic Console** > **Services** > **Messaging** > **JMS Servers**.

    b. Click **Lock & Edit**.

    c. Click **New**.

    Use the following table to create the required JMS Servers. Assign to each JMS Server the previously created persistent stores:

> **Note:**
>
> The number in the names of the existing resources are assigned automatically by the Configuration Wizard during domain creation.
>
> So review the existing JMS server names and select a new and unique name for each new JMS server.
>
> To avoid naming conflicts and simplify the configuration, new resources are qualified with the *product_scaled_N* tag and are shown here as an example.

Cluster to Scale Out	JMS Server Name	Persistent Store	Target
WSM-PM_Cluster	Not applicable	Not applicable	Not applicable
SOA_Cluster	UMSJMSServer_soa_scaled_3	UMSJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	SOAJMSServer_soa_scaled_3	SOAJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	BPMJMSServer_soa_scaled_3	BPMJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	Not applicable	ProcMonJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
OIM_Cluster	OIMJMSServer_scaled_3	JDBCStore-OIM_scaled_3	WLS_OIM3 (migratable)

> **Note:**
>
> (*) BamCQServiceJmsServers host local queues for the BAM CQService (Continuous Query Engine) and are meant to be local. They are intentionally targeted to the WebLogic servers directly and not to the migratable targets.

14. Update the SubDeployment Targets for JMS Modules (if applicable) to include the recently created JMS servers.

    a. Expand the **Services** > **Messaging** > **JMS Modules**.

    b. Click the JMS module. For example: `BPMJMSModule`.

    Use the following table to identify the JMS modules to update, depending on the cluster that you are scaling out:

Table 21-7 The JMS Modules to Update

Cluster to Scale Out	JMS Module to Update	JMS Server to Add to the Subdeployment
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	UMSJMSSystemResource *	UMSJMSServer_soa_scaled_3
	SOAJMSModule	SOAJMSServer_soa_scaled_3
	BPMJMSModule	BPMJMSServer_soa_scaled_3

Table 21-7 (Cont.) The JMS Modules to Update

Cluster to Scale Out	JMS Module to Update	JMS Server to Add to the Subdeployment
	(Only if you have configured Insight) ProcMonJMSModule *	ProcMonJMSServer_soa_scaled_3
OIM_Cluster	OIMJMSModule	OIMJMSServer_scaled_3

(*) Some modules (UMSJMSystemResource, ProcMonJMSModule) may be targeted to more than one cluster. Ensure that you update the appropriate subdeployment in each case.

c. Go to **Configuration** > **Subdeployment**.

d. Add the corresponding JMS Server to the existing subdeployment.

> **Note:**
>
> The subdeployment module name is a random name in the form of `SOAJMSServerXXXXXX`, `UMSJMSServerXXXXXX`, or `BPMJMSServerXXXXXX`, resulting from the Configuration Wizard JMS configuration for the first two servers (`WLS_SOA1` and `WLS_SOA2`).

e. Click **Save**, and then click **Activate Changes.**

15. Start the Node Manager in the Managed Server Domain Directory on OIMHOST3. Follow these steps to update and start the Node Manager from the Managed Server home

    a. Verify that the listen address in the `nodemanager.properties` file is set correctly, by completing the following steps

    - Change directory to the *MSERVER_HOME* binary directory:

      ```
      cd MSERVER_HOME/nodemanager
      ```

    - Open the `nodemanager.properties` file for editing.

    - Validate the **ListenAddress** property to the correct hostname as follows:

      ```
      OIMHOST3: ListenAddress=OIMHOST3
      ```

    - Update the **ListenPort** property with the correct Listen Port details.

    - Make sure that **QuitEnabled** is set to **true**. If this line is not present in the **nodemanager.properties** file, add the following line:

      QuitEnabled=true

    b. Change directory to the *MSERVER_HOME* binary directory:

    ```
    cd MSERVER_HOME /bin
    ```

    c. Use the following command to start the Node Manager:

    nohup ./startNodeManager.sh > $*MSERVER_HOME*/nodemanager/nodemanager.out 2>&1 &

    For information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

16. Restart all servers (except the newly created server) for the previous changes to be effective. You can restart in a rolling manner to eliminate downtime.

17. The configuration is finished. Now sign in to the new host and run the *pack* command to create a template pack, as follows:

    ```
    cd ORACLE_COMMON_HOME/common/bin
    ./pack.sh -managed=true
              -domain=ASERVER_HOME
              -template=/full_path/scaleout_domain.jar
              -template_name=scaleout_domain_template
              -log_priority=DEBUG -log=/tmp/pack.log
    ```

    In this example:

    - Replace *ASERVER_HOME* with the actual path to the domain directory that you created on the shared storage device.
    - Replace *full_path* with the complete path to the location where you want to create the domain template jar file. You need to reference this location when you copy or unpack the domain template jar file. Oracle recommends that you choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

      You must specify a full path for the template jar file as part of the `-template` argument to the `pack` command:

      ```
      SHARED_CONFIG_DIR/domains/template_filename.jar
      ```

    - `scaleout_domain.jar` is a sample name for the jar file that you are creating, which contains the domain configuration files.
    - `scaleout_domain_template` is the label that is assigned to the template data stored in the template file.

18. Run the `unpack` command on `SOAHOSTN` to unpack the template in the managed server domain directory, as follows:

    ```
    cd ORACLE_COMMON_HOME/common/bin
    ./unpack.sh -domain=MSERVER_HOME
                -overwrite_domain=true
                -template=/full_path/scaleout_domain.jar
                -log_priority=DEBUG
                -log=/tmp/unpack.log
                -app_dir=APPLICATION_HOME
    ```

    In this example:

    - Replace *MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain is unpacked.
    - Replace `/full_path/scaleout_domain.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device
    - Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

19. When scaling out the OAM_Cluster:

Register the new Managed Server with Oracle Access Management by doing the following:

a. Log in to the Access Management console at `http://IADADMIN.example.com/oamconsole` as the user you specified during response file creation.

b. Go to the **Configuration** tab.

c. Click **Server Instances**.

d. Select **Create** from the Actions menu.

e. Enter the following information:

   - **Server Name**: `WLS_OAM3`
   - **Host**: Host that the server runs on
   - **Port**: Listen port that was assigned when the Managed Server was created
   - **OAM Proxy Port**: Port you want the Access Manager proxy to run on. This is unique for the host
   - **Proxy Server ID**: `AccessServerConfigProxy`

f. Click **Apply**.

g. Restart the WebLogic Administration Server.

Add the newly created Access Manager server to all of the WebGate Profiles that might be using it, such as `Webgate_IDM`, `Webgate_IDM_11g`, and `IAMSuiteAgent`

For example, to add the Access Manager server to `Webgate_IDM`, access the Access Management console at `http://IADADMIN.example.com/oamconsole`, and do the following:

a. Log in as the Access Manager Administrative User.

b. Go to the **Application Security** tab.

c. Click **Agents**.

d. Click **Search**. You should see the WebGate agent **Webgate_IDM**.

e. Click the agent **Webgate_IDM**.

f. Select **Edit** from the **Actions** menu.

g. Click **+** in the **Primary Server** list (or the **Secondary Server** list if this is a secondary server).

h. Select the newly created managed server from the **Server** list.

i. Set **Maximum Number of Connections** to `10`.

j. Click **Apply**.

k. Repeat the steps for all of the WebGate that are in use, and restart the new Managed Server.

20. Start Node Manager on the new host.

```
cd $NM_HOME
nohup ./startNodeManager.sh > ./nodemanager.out 2>&1 &
```

21. Start the new managed server.

22. Update the web tier configuration to include the new server. When using OHS, there is no need to add the new server to OHS. By default, the Dynamic Server List is used, which means that the list of servers in the cluster is automatically updated when a new node

becomes part of the cluster. So, adding it to the list is not mandatory. The *WebLogicCluster* directive needs only a sufficient number of redundant `server:port` combinations to guarantee the initial contact in case of a partial outage.

If there are expected scenarios where the Oracle HTTP Server is restarted and only the new server is up, update the `WebLogicCluster` directive to include the new server.

For example:

```
<Location /osb>
 WLSRequest ON
 WebLogicCluster SOAHOST1:8011,SOAHOST2:8011,SOAHOST3:8011
 WLProxySSL ON
 WLProxySSLPassThrough ON
</Location>
```

Verifying the Scale Out of Static Clusters

After scaling out and starting the server, proceed with the following verifications:

1. Verify the correct routing to web applications.

   For example:

   a. Access the application on the load balancer:

      ```
      https://igdinternal.example.com:7777/soa-infra
      ```

   b. Check that there is activity in the new server also:

      Go to **Cluster** > **Deployments** > **soa-infra** > **Monitoring** > **Workload**.

   c. You can also verify that the web sessions are created in the new server:

      - Go to **Cluster** > **Deployments**.
      - Expand **soa-infra**, click **soa-infra** Web application.
      - Go to **Monitoring** to check the web sessions in each server.

      You can use the sample URLs and the corresponding web applications that are identified in the following table, to check if the sessions are created in the new server for the cluster that you are scaling out:

Cluster to Verify	Sample URL to Test	Web Application Module
WSM-PM_Cluster	`http://igdinternal.example.com/wsm-pm`	wsm-pm > wsm-pm
SOA_Cluster	`http://igdinternal.example.com:7777/soa-infra`	soa-infra > soa-infra
OAM_Cluster	`http://login.example.com/oam`	
AMA_Cluster	`http://iadadmin.example.com/access`	

Cluster to Verify	Sample URL to Test	Web Application Module
OIM_Cluster	`https://prov.example.com/identity`	

> **Note:**
>
> When validating OAM you will see an OAM server error presented by OAM. This is normal, the test is to show that OAM is being accessed. The error will disappear when appropriate arguments are passed to the server as part of the normal operations.

2. Verify that JMS messages are being produced and consumed to the destinations, and produced and consumed from the destinations, in the three servers.
   a. Go to **JMS Servers**.
   b. Click **JMS Server** > **Monitoring**.
3. Verify the service migration, as described in Validating Automatic Service Migration in Static Clusters.

Scaling in the Topology for Static Clusters

To scale in the topology for a static cluster:

1. Stop the managed server that you want to delete.
2. If you are using automatic service migration, verify that the resources corresponding to that server are not present in the remaining servers before you shrink and scale in the cluster. In case of using *exactly-once migration* policies, stop all the servers
3. Use the OAM Console to remove OAM Servers from the webgate configuration (If removing OAM Servers).
   a. Log in to the Access Management console at `http://iadadmin.example.com/oamconsole` as the user you specified during response file creation.
   b. Go to the **System Configuration** tab.
   c. Click **Server Instances**.
   d. Select the server instances you wish to remove and select **delete** from the actions menu.
   e. Click Apply.
4. Use the Oracle WebLogic Server Administration Console to delete the migratable target that is used by the server that you want to delete.
   a. Click **Lock & Edit**.
   b. Go to **Domain** > **Environment** > **Cluster** > **Migratable Target**.
   c. Select the migratable target that you want to delete.
   d. Click **Delete**.
   e. Click **Yes**.
   f. Click **Activate Changes**.
5. Use the Oracle WebLogic Server Administration Console to delete the new server:

a. Click **Lock & Edit**.

b. Go to **Domain** > **Environment** > **Servers**.

c. Select the server that you want to delete.

d. Click **Delete**.

e. Click **Yes**.

f. Click **Activate Changes**.

> **Note:**
>
> If migratable target was not deleted in the previous step, you get the following error message:
>
> ```
> The following failures occurred: --MigratableTargetMBean WLS_SOA3_soa-
> failure-recovery (migratable) does not have a preferred server set.
> Errors must be corrected before proceeding.
> ```

6. Use the Oracle WebLogic Server Administration Console to update the subdeployment of each JMS Module that is used by the cluster that you are shrinking.

   Use the following table to identify the module for each cluster and perform this action for each module:

Cluster to Scale in	Persistent Store	JMS Server to Delete from the Subdeployment
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	UMSJMSSystemResource SOAJMSModule BPMJMSModule	UMSJMSServer_soa_scaled_3 SOAJMSServer_soa_scaled_3 BPMJMSServer_soa_scaled_3
OIM_Cluster	OIMJMSModule	OIMJMSServer_scaled_3

a. Click **Lock & Edit**.

b. Go to **Domain** > **Services** > **Messaging** > **JMS Modules**.

c. Click the JMS module.

d. Click **subdeployment**.

e. Unselect the JMS server that was created for the deleted server.

f. Click **Save**.

g. Click **Activate Changes**.

7. Use the Oracle WebLogic Server Administration Console to delete the JMS servers:

a. Click **Lock & Edit**.

b. Go to **Domain** > **Services** > **Messaging** > **JMS Servers**.

c. Select the JMS Servers that you created for the new server.

d. Click **Delete**.

e. Click **Yes**.

f. Click **Activate Changes**.

8. Use the Oracle WebLogic Server Administration Console to delete the JMS persistent stores:
    a. Click **Lock & Edit**.
    b. Go to **Domain** > **Services** > **Persistent Stores**.
    c. Select the Persistent Stores that you created for the new server.
    d. Click **Delete**.
    e. Click **Yes**.
    f. Click **Activate Changes**.
9. Update the web tier configuration to remove references to the new server.

Scaling Out the Topology for Dynamic Clusters

This section lists the prerequisites, explains the procedure to scale out the topology with dynamic clusters, describes the steps to verify the scale-out process, and finally the steps to scale down (shrink).

- Prerequisites for Scaling Out
- Scaling Out a Dynamic Cluster
- Verifying the Scale Out of Dynamic Clusters
- Scaling in the Topology for Dynamic Clusters

Prerequisites for Scaling Out

Before you perform a scale out of the topology, you must ensure that you meet the following requirements:

- The starting point is a cluster with managed servers already running.
- The new node can access the existing home directories for WebLogic Server and Governance. Use the existing installations in shared storage. You do not need to install WebLogic Server or IDM binaries in a new location. However, you do need to run `pack` and `unpack` commands to bootstrap the domain configuration in the new node.
- It is assumed that the cluster syntax is used for all internal RMI invocations, JMS adapter, and so on.

Scaling Out a Dynamic Cluster

The steps provided in this procedure use the IAM EDG topology as a reference. Initially there are two application tier hosts (OIMHOST1 and OIMHOST2), each running one managed server of each cluster. A new host OIMHOST3 is added to scale up the clusters with a third managed server. `WLS_XYZ`*`n`* is the generic name given to the new managed server that you add to the cluster. Depending on the cluster that is being extended and the number of existing nodes, the actual names are `WLS_SOA3` and `WLS_OIM3`.
To scale out the topology in a dynamic cluster, complete the following steps:

1. On the new node, mount the existing shared volumes for FMW Home (Binaries1), shared config (sharedConfig), and runtime (runTime), as described in Summary of the Shared Storage Volumes in an Enterprise Deployment

> **Note:**
>
> Be sure to mount the file systems associated with the Identity Governance binaries.

2. Locate the inventory in the shared directory (for example, `/u01/oracle/products/oraInventory`), per Oracle's recommendation. So you do not need to attach any home, but you may want to execute the script: `/u01/oracle/products/oraInventory/createCentralInventory.sh`.

   This command creates and updates the local file `/etc/oraInst.loc` in the new node to point it to the `oraInventory` location.

   If there are other inventory locations in the new host, you can still use them, but `/etc/oraInst.loc` file must be updated accordingly for updates in each case.

3. Update the `/etc/hosts` files to add the name of the new host (unless you are using DNS), as described in Verifying IP Addresses and Host Names in DNS or Hosts File. If you are using host aliases such as OIMHOST, then ensure that you add an entry for that host.

   For example:

   ```
   10.229.188.204 host1-vip.example.com host1-vip ADMINVHN
   10.229.188.205 host1.example.com host1 OIMHOST1
   10.229.188.206 host2.example.com host2 OIMHOST2
   10.229.188.207 host3.example.com host3 WEBHOST1
   10.229.188.208 host4.example.com host4 WEBHOST2
   10.229.188.209 host5.example.com host5 OIMHOST3
   ```

4. Configure a per domain Node Manager in the new node, as described in Creating a Per Host Node Manager Configuration.

5. Log in to the Oracle WebLogic Administration Console to create a new machine for the new node.

6. Update the machine's Node Manager address to map the IP of the node that is being used for scale out.

7. Use the Oracle WebLogic Server Administration Console to increase the dynamic cluster to include a new managed server:

   a. Click **Lock & Edit**.

   b. Go to **Domain** > **Environment** > **Clusters**.

   c. Select the cluster to want to scale out.

   d. Go to **Configuration** > **Servers**.

   e. Set **Dynamic Cluster Size** to 3. By default, the cluster size is 2.

   > **Note:**
   >
   > In case of scaling-out to more than three servers, we also need to update *Number of servers in cluster Address* that is 3 by default. Although Oracle recommends you to use the cluster syntax for t3 calls, the cluster address is used if calling from external elements via t3, for EJB stubs, and so on.

8. Sign in to OIMHOST1 and run the *pack* command to create a template pack as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin
   ./pack.sh -managed=true
             -domain=ASERVER_HOME
             -template=/full_path/scaleout_domain.jar
             -template_name=scaleout_domain_template
             -log_priority=DEBUG -log=/tmp/pack.log
   ```

   In this example:

   - Replace *ASERVER_HOME* with the actual path to the domain directory that you created on the shared storage device.
   - Replace *full_path* with the complete path to the location where you want to create the domain template jar file. You need to reference this location when you copy or unpack the domain template jar file. Oracle recommends that you choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

     You must specify a full path for the template jar file as part of the `-template` argument to the `pack` command:

     ```
     SHARED_CONFIG_DIR/domains/template_filename.jar
     ```

   - `scaleout_domain.jar` is a sample name for the jar file that you are creating, which contains the domain configuration files.
   - `scaleout_domain_template` is the label that is assigned to the template data stored in the template file.

9. Run the `unpack` command on `SOAHOSTN` to unpack the template in the managed server domain directory, as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin
   ./unpack.sh -domain=MSERVER_HOME
               -overwrite_domain=true
               -template=/full_path/scaleout_domain.jar
               -log_priority=DEBUG
               -log=/tmp/unpack.log
               -app_dir=APPLICATION_HOME
   ```

   In this example:

   - Replace *MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain is unpacked.
   - Replace `/full_path/scaleout_domain.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device
   - Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

10. Start Node Manager on the new host.

    ```
    cd $NM_HOME
    nohup ./startNodeManager.sh > ./nodemanager.out 2>&1 &
    ```

11. Start the new managed Server.
12. Update the web tier configuration to include this new server. If using OHS, there is no need to add the new server to OHS.

    By default, the Dynamic Server list is used, which means that the list of servers in the cluster is automatically updated when a new node becomes part of the cluster. So adding the new node to the list is not mandatory. The *WebLogicCluster* directive needs only a sufficient number of redundant `server:port` combinations to guarantee the initial contact in the case of a partial outage.

    If there expected scenarios where the Oracle HTTP Server is restarted and only the new server would be up, update the WebLogicCluster directive to include the new server.

    For example:

    ```
    <Location /soa-infra>
     WLSRequest ON
     WebLogicCluster OIMHOST1:8011,OIMHOST2:8012,OIMHOST3:8013
     WLProxySSL ON
     WLProxySSLPassThrough ON
    </Location>
    ```

    - If using OHS, there is no need to add the new server to OHS.

      By default, the Dynamic Server list is used, which means that the list of servers in the cluster is automatically updated when a new node becomes part of the cluster. So adding the new node to the list is not mandatory. The *WebLogicCluster* directive needs only a sufficient number of redundant `server:port` combinations to guarantee the initial contact in the case of a partial outage.

      If there expected scenarios where the Oracle HTTP Server is restarted and only the new server would be up, update the WebLogicCluster directive to include the new server.

      For example:

      ```
      <Location /soa-infra>
       WLSRequest ON
       WebLogicCluster OIMHOST1:8011,OIMHOST2:8012,OIMHOST3:8013
       WLProxySSL ON
       WLProxySSLPassThrough ON
      </Location>
      ```

Verifying the Scale Out of Dynamic Clusters

After scaling out and starting the server, proceed with the following verifications:

1. Verify the correct routing to web applications.

   For example:

a. Access the application on the load balancer:

```
https://igdinternal.example.com:7777/soa-infra
```

b. Check that there is activity in the new server also:

Go to **Cluster** > **Deployments** > **soa-infra** > **Monitoring** > **Workload**.

c. You can also verify that the web sessions are created in the new server:

- Go to **Cluster** > **Deployments**.
- Expand **soa-infra**, click **soa-infra** Web application.
- Go to **Monitoring** to check the web sessions in each server.

You can use the sample URLs and the corresponding web applications that are identified in the following table, to check if the sessions are created in the new server for the cluster that you are scaling out:

Cluster to Verify	Sample URL to Test	Web Application Module
WSM-PM_Cluster	`http://igdinternal.example.com/wsm-pm`	wsm-pm > wsm-pm
SOA_Cluster	`http://igdinternal.example.com:7777/soa-infra`	soa-infra > soa-infra
OAM_Cluster	`http://login.example.com/oam`	
AMA_Cluster	`http://iadadmin.example.com/access`	
OIM_Cluster	`https://prov.example.com/identity`	

Note:

When validating OAM you will see an OAM server error presented by OAM. This is normal, the test is to show that OAM is being accessed. The error will disappear when appropriate arguments are passed to the server as part of the normal operations.

2. Verify that JMS messages are being produced and consumed to the destinations, and produced and consumed from the destinations, in the three servers.

   a. Go to **JMS Servers**.

   b. Click **JMS Server** > **Monitoring**.

3. Verify the service migration, as described in Validating Automatic Service Migration in Dynamic Clusters.

Scaling in the Topology for Dynamic Clusters

To scale in the topology for a dynamic cluster:

1. Stop the managed server that you want to delete.
2. If you are using automatic service migration, verify that the singleton resources corresponding to that server are not present in the remaining servers before you shrink/ scale in the cluster.
3. Use the Oracle WebLogic Server Administration Console to reduce the dynamic cluster:
    a. Click **Lock & Edit**.
    b. Go to **Domain** > **Environment** > **Clusters**.
    c. Select the cluster that you want to scale in.
    d. Go to **Configuration** > **Servers**.
    e. Set the Dynamic Cluster size to `2`.
4. If you are using OSB, restart the Admin Server.

Scaling Up the Topology

When you scale up the topology, you add new managed servers to the existing hosts.

This section describes the procedures to scale up the topology with static and dynamic clusters.

- Scaling Up the Topology for Static Clusters
  This section lists the prerequisites, explains the procedure to scale up the topology with static clusters, describes the steps to verify the scale-out process, and finally the steps to scale down (shrink).
- Scaling Up the Topology for Dynamic Clusters
  This section lists the prerequisites, explains the procedure to scale out the topology with dynamic clusters, describes the steps to verify the scale-up process, and finally the steps to scale down (shrink).

Scaling Up the Topology for Static Clusters

This section lists the prerequisites, explains the procedure to scale up the topology with static clusters, describes the steps to verify the scale-out process, and finally the steps to scale down (shrink).

You already have a node that runs a managed server that is configured with Fusion Middleware components. The node contains a WebLogic Server home and an Oracle Fusion Middleware IAMhome in shared storage. Use these existing installations and domain directories, to create the new managed servers. You do not need to install WLS or SOA binaries or to run *pack* and *unpack* because the new server is going to run in the existing node.

- Prerequisites for Scaling Up
- Scaling Up a Static Cluster
- Verifying the Scale Up of Static Clusters
- Scaling in the Topology for Static Clusters

Prerequisites for Scaling Up

Before you perform a scale up of the topology, you must ensure that you meet the following requirements:

- The starting point is a cluster with managed servers already running.
- It is assumed that the cluster syntax is used for all internal RMI invocations, JMS adapter, and so on.

Scaling Up a Static Cluster

The IAM EDG topology has two different domains, one for OAM and one for OIG. Scaling Up is largely the same regardless of which cluster you are scaling. The example below refers to HOST1, HOST2 and HOST3. If you are scaling OAM then these hosts will equate to OAMHOST1, OAMHOST2 and OAMHOST3. If you are scaling OIM then these hosts will equate to OIMHOST1, OIMHOST2 and OIMHOST3.

The example below explains how to add a third managed server to the cluster that runs in HOST1. WLS_XYZn is the generic name given to the new managed server that you add to the cluster. Depending on the cluster that is being extended and the number of existing nodes, the actual names are `WLS_OAM3`, `WLS_AMA3`, `WLS_OIM4`, `WLS_SOA3`, `WLS_WSM3`, and so on.

To scale up the cluster, complete the following steps:

1. Use the Oracle WebLogic Server Administration Console to clone the first managed server in the cluster into a new managed server.
   a. In the Change Center section, click **Lock & Edit**.
   b. Go to **Environment** > **Servers**.
   c. Select the first managed server in the cluster to scale up and click **Clone**.
   d. Use Table 21-8 to set the correspondent name, listen address, and listen port depending on the cluster that you want to scale out. Note that the default listen port is increment by 1 to avoid binding conflicts with the managed server that is already created and running in the same host.
   e. Click the new managed server, and then select **Configuration** > **General**.
   f. Click **Save**, and then click **Activate Changes**.

Table 21-8 List of Clusters that You Want to Scale Up

Cluster to Scale Up	Server to Clone	New Server Name	Server Listen Address	Server Listen Port
WSM-PM_Cluster	WLS_WSM1	WLS_WSM*n*	OIMHOST*n*	7011
SOA_Cluster	WLS_SOA1	WLS_SOA*n*	OIMHOST*n*	8001
OIM_Cluster	WLS_OIM1	WLS_OIM*n*	OIMHOST*n*	14000
OAM_Cluster	WLS_OAM1	WLS_OAM*n*	OAMHOST*n*	14100
AMA_Cluster	WLS_AMA1	WLS_AMA*n*	OAMHOST*n*	14150

Note:

Port numbers must be unique on a given host, therefore the port numbers above have been incremented by 1 from the ports used by the existing managed servers on the host.

2. Update the deployment Staging Directory Name of the new server, as described in Modifying the Upload and Stage Directories to an Absolute Path.

3. Create a new key certificate and update the private key alias of the server, as described in Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer.
4. By default, the cloned server uses default store for TLOGs. If the rest of the servers in the cluster that you are scaling-out are using TLOGs in JDBC persistent store, update the TLOG persistent store of the new managed server:

   Use the following table to identify the clusters that use JDBC TLOGs by default:

   Table 21-9 The Name of Clusters that Use JDBC TLOGs by Default

Cluster to Scale Up	New Server Name	TLOG Persistent Store
WSM-PM_Cluster	WLS_WSM*n*	Default (file)
SOA_Cluster	WLS_SOA*n*	JDBC
OIM_Cluster	WLS_OIM*n*	Default JDBC
OAM_Cluster	WLS_OAM*n*	Not Applicable
AMA_Cluster	WLS_AMA*n*	Not Applicable

   Complete the following steps

   a. Go to **Environment** > **Servers** > **WLS_XYZn** > **Configuration** > **Services**.
   b. In **Transaction Log Store** section, change **Type** to JDBC.
   c. Change **Data Source** to *WLSSchemaDatasource*.
   d. Click **Save**, and then click **Activate Changes**.
5. If the cluster you are scaling up is configured for automatic service migration, update the **JTA Migration Policy** to the required value.

   Use the following table to identify the clusters for which you have to update the JTA Migration Policy:

   Table 21-10 The Recommended JTA Migration Policy for the Cluster to be Scaled Up

Cluster to Scale Up	New Server Name	JTA Migration Policy
WSM-PM_Cluster	WLS_WSM*n*	Manual
SOA_Cluster	WLS_SOA*n*	Failure Recovery
OAM_Cluster	WLS_OAM*n*	Not Applicable
AMA_Cluster	WLS_AMA*n*	Not Applicable
OIM_Cluster	WLS_OIM*n*	Failure Recovery

   Complete the following steps:

   a. Go to **Environment** > **Servers** > **WLS_XYZn** > **Configuration** > **Migration**.
   b. Use Table 21-10 to set the recommended JTA Migration Policy depending on the cluster that you want to scale out.
   c. Click **Save**, and then click **Activate Changes**.
   d. For the rest of the servers already existing in the cluster, update the list of **JTA candidate servers** for JTA migration to include the new server.
      - Go to **Environment** > **Servers** > **server** > **Configuration** > **Migration**.

- Go to **JTA Candidate Servers**: leave the list empty (leave it empty because all servers in the cluster are JTA candidate servers).
- Click **Save**, and then click **Activate Changes**. Although you need to restart the servers for this change to be effective, you can do a unique restart later, after you complete all the required configuration changes.

6. If the cluster you are scaling up is configured for automatic service migration, use the Oracle WebLogic Server Administration Console to update the automatically created WLS_XYZn (migratable) with the recommended migration policy, because by default it is set to **Manual Service Migration Only**.

   Use the following table for the list of migratable targets to update:

   Table 21-11 The Recommended Migratable Targets to Update

Cluster to Scale Up	Migratable Target to Update	Migration Policy
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	WLS_SOA*n* (migratable)	Auto-Migrate Failure Recovery Services
OIM_Cluster	WLS_OIM*n* (migratable)	Auto-Migrate Failure Recovery Services
OAM_Cluster	Not applicable	Not applicable
AMA_Cluster	Not applicable	Not applicable

   a. Go to **Environment** > **Cluster** > **Migratable Servers**.
   b. Click **Lock and Edit**.
   c. Click WLS_XYZ3 (migratable).
   d. Go to the **Configuration** tab and then **Migration**.
   e. Change the **Service Migration Policy** to the value listed in the table.
   f. Leave the **Constrained Candidate Server** list blank in case there are chosen servers. If no servers are selected, you can migrate this migratable target to any server in the cluster.
   g. Click **Save**, and then click **Activate Changes**.

7. Update the **Constrained Candidate Server** list in the existing migratable servers in the cluster that you are scaling because by default they are pre-populated with only WLS_XYZ1 and WLS_XYZ2 servers.

   Use the following table to identify the migratable servers that have to be updated:

   Table 21-12 The Existing Migratable Targets to Update

Cluster to Scale Up	Existing Migratable Target to Update	Constrained Candidate Server
WSM-PM_Cluster	Not applicable	Leave empty
SOA_Cluster	WLS_SOA1 (migratable) WLS_SOA2 (migratable)	Leave empty
OIM_Cluster	WLS_OIM1 (migratable) WLS_OIM1 (migratable)	Leave empty

   a. Go to each migratable server.
   b. Go to the tab **Configuration** > **Migration** > **Constrained Candidate Server**.

      You can leave the server list blank to make these migratable targets migrate to any server in this cluster, including the newly created managed server.
   c. Click **Save** and **Activate Changes**. Although you need to restart the servers for this change to be effective, you can do a unique restart later, after you complete all the required configuration changes.
8. Create the required persistent stores for the JMS servers.
   a. Sign in to WebLogic Console and go to **Services** > **Persistent Stores**.
   b. Click **New** and select **Create JDBCStore**.

   Use the following table to create the required persistent stores:

> **Note:**
>
> The number in the names and prefixes in the existing resources were assigned automatically by the Configuration Wizard during the domain creation.
>
> For example:
>
> ```
> BPMJMSJDBCStore_auto_1 — soa_1
> BPMJMSJDBCStore_auto_2 — soa_2
> JDBCStore-OIM_auto_1 - oim1
> JDBCStore-OIM_auto_2 - oim2
> SOAJMSJDBCStore_auto_1 - soa_1
> SOAJMSJDBCStore_auto_2 - soa_2
> UMSJMSJDBCStore_auto_1 - soa_1
> UMSJMSJDBCStore_auto_2 - soa_2
> ```
>
> Review the existing prefixes and select a new and unique prefix and name for each new persistent store.
>
> To avoid naming conflicts and simplify the configuration, new resources are qualified with the *scaled* tag and are shown here as an example.

Table 21-13 The New Resources Qualified with the Scaled Tag

Cluster to Scale Up	Persistent Store	Prefix Name	Data Source	Target
WSM-PM_Cluster	Not applicable	Not applicable	Not applicable	Not applicable
SOA_Cluster	UMSJMSJDBCStore_soa_scaled_3	soaums_scaled_3	WLSSchemaData Sourc	WLS_SOA3 (migratable)
	SOAJMSJDBCStore_ soa_scaled_3	soajms_scaled_3	WLSSchemaData Sourc	WLS_SOA3 (migratable)
	BPMJMSJDBCStore_ soa_scaled_3	soabpm_scaled_3	WLSSchemaData Sourc	WLS_SOA3 (migratable)

Table 21-13 (Cont.) The New Resources Qualified with the Scaled Tag

Cluster to Scale Up	Persistent Store	Prefix Name	Data Source	Target
	(only when you use Insight) ProcMonJMSJDBCStore_soa_scaled_3	soaprocmon_scaled_3	WLSSchemaDataSource	WLS_SOA3 (migratable)
OIM_Cluster	JDBCStore-OIM_scaled_3	oimjms_scaled_3	WLSSchemaDataSource	WLS_OIM3 (migratable)

9. Create the required JMS Servers for the new managed server.

   a. Go to **WebLogic Console** > **Services** > **Messaging** > **JMS Servers**.

   b. Click **Lock and Edit**.

   c. Click **New**.

   Use the following table to create the required JMS Servers. Assign to each JMS Server the previously created persistent stores:

   > **Note:**
   >
   > The number in the names of the existing resources are assigned automatically by the Configuration Wizard during domain creation. Review the existing JMS server names and select a new and unique name for each new JMS server. To avoid naming conflicts and simplify the configuration, new resources are qualified with the *product_scaled_N* tag and are shown here as an example.

Cluster to Scale Up	JMS Server Name	Persistent Store	Target
WSM-PM_Cluster	Not applicable	Not applicable	Not applicable
SOA_Cluster	UMSJMSServer_soa_scaled_3	UMSJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	SOAJMSServer_soa_scaled_3	SOAJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	BPMJMSServer_soa_scaled_3	BPMJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	(only when you use Insight) ProcMonJMSServer_soa_scaled_3	ProcMonJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
OIM_Cluster	OIMJMSServer_scaled_3	JDBCStore-OIM_scaled_3	WLS_OIM3 (migratable)

10. Update the SubDeployment Targets for JMS Modules (if applicable) to include the recently created JMS servers.

    a. Expand the **Services**>**Messaging**>**JMS Modules**.

    b. Click the JMS module. For example: `BPMJMSModule`.

       Use the following table to identify the JMS modules to update depending on the cluster that you are scaling up:

Cluster to Scale-up	JMS Module to Update	JMS Server to Add to the Subdeployment
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	UMSJMSSystemResource *	UMSJMSServer_soa_scaled_3
	SOAJMSModule	SOAJMSServer_soa_scaled_3
	BPMJMSModule	BPMJMSServer_soa_scaled_3
	(Only if you have configured Insight) ProcMonJMSModule *	ProcMonJMSServer_soa_scaled_3
OIM_Cluster	OIMJMSModule	OIMJMSServer_scaled_3

(*) Some modules (UMSJMSystemResource, ProcMonJMSModule) may be targeted to more than one cluster. Ensure that you update the appropriate subdeployment in each case.

c. Go to **Configuration** > **Subdeployment**.

d. Add the corresponding JMS Server to the existing subdeployment.

> **Note:**
>
> The Subdeployment module name is a random name in the form of `SOAJMSServerXXXXXX`, `UMSJMSServerXXXXXX`, or `BPMJMSServerXXXXXX`, resulting from the Configuration Wizard JMS configuration for the first two servers (`WLS_SOA1` and `WLS_SOA2`).

e. Click **Save**, and then click **Activate Changes**.

11. Restart all servers (except the newly created server) for all the previous changes to be effective. You can restart in a rolling manner to eliminate downtime.

12. Start the new managed server.

13. Update the web tier configuration to include this new server. When using OHS, there is no need to add the new server to OHS. By default Dynamic Server List is used, which means that the list of the servers in the cluster is automatically updated when a new node become part of the cluster, so adding it to the list is not mandatory. The WebLogicCluster directive needs only a sufficient number of redundant `server:port` combinations to guarantee initial contact in case of a partial outage.

    If there are expected scenarios where the Oracle HTTP Server is restarted and only the new server would be up, update the WebLogicCluster directive to include the new server.

```
<Location /soa-infra>
  WLSRequest ON
  WebLogicCluster OIMHOST1:8011,OIMHOST2:8012,OIMHOST3:8013
  WLProxySSL ON
  WLProxySSLPassThrough ON
</Location>
```

Verifying the Scale Up of Static Clusters

After scaling up and starting the server, proceed with the following verifications:

1. Verify the correct routing to web applications.

   For example:

   a. Access the application on the load balancer:

      ```
      https://igdinternal.example.com:7777/soa-infra
      ```

   b. Check that there is activity in the new server also:

      Go to **Cluster** > **Deployments** > **soa-infra** > **Monitoring** > **Workload**.

   c. You can also verify that the web sessions are created in the new server:

      - Go to **Cluster** > **Deployments**.
      - Expand **soa-infra**, click **soa-infra** Web application.
      - Go to **Monitoring** to check the web sessions in each server.

      You can use the sample URLs and the corresponding web applications that are identified in the following table, to check if the sessions are created in the new server for the cluster that you are scaling out:

Cluster to Verify	Sample URL to Test	Web Application Module
WSM-PM_Cluster	`http://igdinternal.example.com/wsm-pm`	wsm-pm > wsm-pm
SOA_Cluster	`http://igdinternal.example.com:7777/soa-infra`	soa-infra > soa-infra
OAM_Cluster	`http://login.example.com/oam`	
AMA_Cluster	`http://iadadmin.example.com/access`	
OIM_Cluster	`https://prov.example.com/identity`	

> **Note:**
>
> When validating OAM you will see an OAM server error presented by OAM. This is normal, the test is to show that OAM is being accessed. The error will disappear when appropriate arguments are passed to the server as part of the normal operations.

2. Verify that JMS messages are being produced and consumed to the destinations, and produced and consumed from the destinations, in the three servers.

   a. Go to **JMS Servers**.

   b. Click **JMS Server** > **Monitoring**.

3. Verify the service migration, as described in Validating Automatic Service Migration in Static Clusters.

Scaling in the Topology for Static Clusters

To scale in the topology for a static cluster:

1. Stop the managed server that you want to delete.
2. If you are using automatic service migration, verify that the resources corresponding to that server are not present in the remaining servers before you shrink and scale in the cluster. In case of using *exactly-once migration* policies, stop all the servers
3. Use the OAM Console to remove OAM Servers from the webgate configuration (If removing OAM Servers).
    1. Log in to the Access Management console at `http://iadadmin.example.com/oamconsole` as the user you specified during response file creation.
    2. Go to the **System Configuration** tab.
    3. Click **Server Instances**.
    4. Select the server instances you wish to remove and select **delete** from the actions menu.
    5. Click Apply.
4. Use the Oracle WebLogic Server Administration Console to delete the migratable target that is used by the server that you want to delete.
    1. Click **Lock & Edit**.
    2. Go to **Domain** > **Environment** > **Cluster** > **Migratable Target**.
    3. Select the migratable target that you want to delete.
    4. Click **Delete**.
    5. Click **Yes**.
    6. Click **Activate Changes**.
5. Use the Oracle WebLogic Server Administration Console to delete the new server:
    1. Click **Lock & Edit**.
    2. Go to **Domain** > **Environment** > **Servers**.
    3. Select the server that you want to delete.
    4. Click **Delete**.
    5. Click **Yes**.
    6. Click **Activate Changes**.

> **Note:**
>
> If migratable target was not deleted in the previous step, you get the following error message:
>
> ```
> The following failures occurred: --MigratableTargetMBean WLS_SOA3_soa-
> failure-recovery (migratable) does not have a preferred server set.
> Errors must be corrected before proceeding.
> ```

6. Use the Oracle WebLogic Server Administration Console to update the subdeployment of each JMS Module that is used by the cluster that you are shrinking.

Use the following table to identify the module for each cluster and perform this action for each module:

Cluster to Scale in	Persistent Store	JMS Server to Delete from the Subdeployment
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	UMSJMSSystemResource SOAJMSModule BPMJMSModule	UMSJMSServer_soa_scaled_3 SOAJMSServer_soa_scaled_3 BPMJMSServer_soa_scaled_3
OIM_Cluster	OIMJMSModule	OIMJMSServer_scaled_3

a. Click **Lock & Edit**.
b. Go to **Domain** > **Services** > **Messaging** > **JMS Modules**.
c. Click the JMS module.
d. Click **subdeployment**.
e. Unselect the JMS server that was created for the deleted server.
f. Click **Save**.
g. Click **Activate Changes**.

7. Use the Oracle WebLogic Server Administration Console to delete the JMS servers:
   a. Click **Lock & Edit**.
   b. Go to **Domain** > **Services** > **Messaging** > **JMS Servers**.
   c. Select the JMS Servers that you created for the new server.
   d. Click **Delete**.
   e. Click **Yes**.
   f. Click **Activate Changes**.
8. Use the Oracle WebLogic Server Administration Console to delete the JMS persistent stores:
   a. Click **Lock & Edit**.
   b. Go to **Domain** > **Services** > **Persistent Stores**.
   c. Select the Persistent Stores that you created for the new server.
   d. Click **Delete**.
   e. Click **Yes**.
   f. Click **Activate Changes**.
9. Update the web tier configuration to remove references to the new server.

Scaling Up the Topology for Dynamic Clusters

This section lists the prerequisites, explains the procedure to scale out the topology with dynamic clusters, describes the steps to verify the scale-up process, and finally the steps to scale down (shrink).

You already have a node that runs a managed server that is configured with Fusion Middleware components. The node contains a WebLogic Server home and an Oracle Fusion Middleware IAM home in shared storage. Use these existing installations and domain directories, to create the new managed servers. You do not need to install WLS or SOA binaries or to run `pack` and `unpack` commands, because the new server is going to run in the existing node.

- Prerequisites for Scaling Up
- Scaling Up a Dynamic Cluster
- Verifying the Scale Up of Dynamic Clusters
- Scaling Down the Topology in a Dynamic Cluster

Prerequisites for Scaling Up

Before performing a scale up of the topology, you must ensure that you meet the following prerequisites:

- The starting point is a cluster with managed servers already running.
- It is assumed that the cluster syntax is used for all internal RMI invocations, JMS adapter, and so on.

Scaling Up a Dynamic Cluster

Use the IAM EDG topology as a reference, with two application tier hosts (OIMHOST1 and OIMHOST2), each running one managed server of each cluster. The example explains how to add a third managed server to the cluster that runs in OIMHOST1. `WLS_XYZ`*`n`* is the generic name given to the new managed server that you add to the cluster. Depending on the cluster that is being extended and the number of existing nodes, the actual names will be `WLS_SOA3` and `WLS_OIM3`.

To scale up the cluster, complete the following steps:

1. In scale-up, there is no need of adding a new machine to the domain as the new server would be added to an existing machine.

   If the *CalculatedMachineNames* attribute is set to true, then the *MachineNameMatchExpression* attribute is used to select the set of machines used for the dynamic servers. Assignments are made by using a round-robin algorithm.

   This following table lists examples of machine assignments in a dynamic cluster.

   Table 21-14 Examples of machine assignments in a dynamic cluster

Machines in Domain	*MachineNameMatchExpression* Configuration	Dynamic Server Machine Assignments
OIMHOST1, OIMHOST2	OIMHOST*	dyn-server-1: OIMHOST1 dyn-server-2: OIMHOST2 dyn-server-3: OIMHOST1 dyn-server-4: OIMHOST2 ...

   See https://docs.oracle.com/middleware/1212/wls/CLUST/dynamic_clusters.htm#CLUST678.

2. If you are using *OIMHOST{$id}* as listen address in the template, update the `/etc/hosts` files to add the alias `SOAHOSTN` for the new node as described in the Verifying IP Addresses and Host Names in DNS or Hosts File.

The new server `WLS_XYZn listens` in `SOAHOSTn`. This alias must be resolved to the corresponding IP address of the system host where the new managed server runs. See Table 21-14.

Example:

```
10.229.188.204 host1-vip.example.com host1-vip ADMINVHN
10.229.188.205 host1.example.com host1 OIMHOST1
10.229.188.206 host2.example.com host2 OIMHOST2
10.229.188.207 host3.example.com host3 WEBHOST1
10.229.188.208 host4.example.com host4 WEBHOST2
10.229.188.209 host5.example.com host5 OIMHOST3
```

If you are using {$dynamic-hostname} in the listen address of the template, the new server `WLS_xYZn` listens in the address defined for the JAVA property `dynamic-hostname` . In this case, adding aliases to `/etc/hosts` file is not necessary when you scale up the dynamic cluster. See Configuring Listen Addresses in Dynamic Cluster Server Templates.

3. Use the Oracle WebLogic Server Administration Console to increase the dynamic cluster to include a new managed server:

   a. Click **Lock & Edit**.

   b. Go to **Domain** > **Environment** > **Clusters**.

   c. Select the cluster to want to scale out.

   d. Go to **Configuration** > **Servers**.

   e. Set **Dynamic Cluster Size** to `3`. By default, the cluster size is 2.

   f. Click **Save**and then, click **Activate Changes**.

   > **Note:**
   >
   > In case of scaling-out to more than three servers, we also need to update *Number of servers in cluster Address* that is 3 by default. Although, Oracle recommends that you use the cluster syntax for t3 calls, the cluster address is used if calling from external elements through t3, for EJB stubs, and so on.

4. Update the web tier configuration to include this new server. When using OHS, there is no need to add the new server to OHS. By default Dynamic Server List is used, which means that the list of the servers in the cluster is automatically updated when a new node become part of the cluster, so adding it to the list is not mandatory. The WebLogicCluster directive needs only a sufficient number of redundant `server:port` combinations to guarantee initial contact in case of a partial outage.

   If there are expected scenarios where the Oracle HTTP Server is restarted and only the new server would be up, update the WebLogicCluster directive to include the new server.

   For example:

```
<Location /soa-infra>
  WLSRequest ON
  WebLogicCluster OIMHOST1:8011,OIMHOST2:8012,OIMHOST3:8013
  WLProxySSL ON
  WLProxySSLPassThrough ON
</Location>
```

5. Start the new managed server from the Oracle WebLogic Server.
6. Verify that the newly created managed server is running.

Verifying the Scale Up of Dynamic Clusters

After you scale out and start the server, proceed with the following verifications:

1. Verify the correct routing to web applications.

   For example:

   a. Access the application on the load balancer:

      ```
      https://igdinternal.example.com:7777/soa-infra
      ```

   b. Check that there is activity in the new server also:

      Go to **Cluster** > **Deployments** > **soa-infra** > **Monitoring** > **Workload**.

   c. You can also verify that the web sessions are created in the new server:

      - Go to **Cluster** > **Deployments**.
      - Expand **soa-infra**, click **soa-infra** Web application.
      - Go to **Monitoring** to check the web sessions in each server.

      You can use the sample URLs and the corresponding web applications that are identified in the following table, to check if the sessions are created in the new server for the cluster that you are scaling out:

Cluster to Verify	Sample URL to Test	Web Application Module
WSM-PM_Cluster	`http://igdinternal.example.com/wsm-pm`	wsm-pm > wsm-pm
SOA_Cluster	`http://igdinternal.example.com:7777/soa-infra`	soa-infra > soa-infra
OAM_Cluster	`http://login.example.com/oam`	
AMA_Cluster	`http://iadadmin.example.com/access`	
OIM_Cluster	`https://prov.example.com/identity`	

> **Note:**
>
> When validating OAM you will see an OAM server error presented by OAM. This is normal, the test is to show that OAM is being accessed. The error will disappear when appropriate arguments are passed to the server as part of the normal operations.

2. Verify that JMS messages are being produced and consumed to the destinations, and produced and consumed from the destinations, in the three servers.

   a. Go to **JMS Servers**.
   b. Click **JMS Server** > **Monitoring**.

3. Verify the service migration, as described in Configuring Automatic Service Migration for Dynamic Clusters.

Scaling Down the Topology in a Dynamic Cluster

To scale down the topology in a dynamic cluster:

1. Stop the managed server that you want to delete.
2. If you are using Automatic Service Migration, verify that the singleton resources corresponding to that server are not present in the remaining servers before you scale down the cluster.
3. Use the Oracle WebLogic Server Administration Console to reduce the dynamic cluster:
   a. Click **Lock & Edit**.
   b. Go to **Domain** > **Environment** > **Clusters**.
   c. Select the cluster to want to scale-down.
   d. Go to **Configuration** > **Servers**.
   e. Set again the **Dynamic Cluster Size** to `2`.

OAM Specific Scaling Actions

This section briefs about registering any new managed servers with access manager and Webgate Agents.

In addition to the steps above to scale up or out the OAM managed servers. You must also register any new managed servers with Access Manager and Webgate Agents. To do this you need to perform the following steps.

- Register new OAM Managed Servers
- Updating WebGate Profiles

Register new OAM Managed Servers

Register the new Managed Server with Oracle Access Management Access Manager. You now must configure the new Managed Server now as an Access Manager server. You do this from the Oracle Access Management Console. Proceed as follows:

1. Log in to the Access Management console at `http://IADADMIN.example.com/oamconsole` as the user you specified during response file creation.
2. Click the **System Configuration** tab.
3. Click **Server Instances**.
4. Select **Create** from the Actions menu.
5. Enter the following information:
   - **Server Name**: `WLS_OAM3`
   - **Host**: Host that the server runs on
   - **Port**: Listen port that was assigned when the Managed Server was created

- **OAM Proxy Port**: Port you want the Access Manager proxy to run on. This is unique for the host
- **Proxy Server ID**: `AccessServerConfigProxy`
- **Mode**: Set to same mode as existing Access Manager servers.

6. Click **Coherence** tab.

   Set **Local Port** to a unique value on the host.
7. Click **Apply**.
8. Restart the WebLogic Administration Server.

Updating WebGate Profiles

Add the newly created Access Manager server to all WebGate Profiles that might be using it, such as `Webgate_IDM`, `Webgate_IDM_12c`, and `IAMSuiteAgent`

For example, to add the Access Manager server to `Webgate_IDM`, access the Access Management console at: `http://IADADMIN.example.com/oamconsole`

Then proceed as follows:

1. Log in as the Access Manager Administrative User.
2. Click the **System Configuration** tab.
3. Expand **Access Manager Settings - SSO Agents - OAM Agents**.
4. Click the open folder icon, then click **Search**.

   You should see the WebGate agent **Webgate_IDM**.
5. Click the agent **Webgate_IDM**.
6. Select **Edit** from the **Actions** menu.
7. Click **+** in the **Primary Server** list (or the **Secondary Server** list if this is a secondary server).
8. Select the newly created managed server from the **Server** list.
9. Set **Maximum Number of Connections** to `10`.
10. Click **Apply**.

Repeat Steps 5 through 10 for **Webgate_IDM_12c**, **IAMSuiteAgent**, and all other WebGates that might be in use.

22 Configuring Single Sign-On for an Enterprise Deployment

You need to configure the Oracle HTTP Server WebGate in order to enable single sign-on with Oracle Access Manager.

- About Oracle Webgate
  Oracle WebGate is a web server plug-in that intercepts HTTP requests and forwards them to an existing Oracle Access Manager instance for authentication and authorization.
- General Prerequisites for Configuring Oracle HTTP Server WebGate
  Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.
- Configuring Oracle HTTP Server 12c WebGate for an Enterprise Deployment
  You need to perform the following steps in order to configure Oracle HTTP Server 12*c* WebGate for Oracle Access Manager on both WEBHOST1 and WEBHOST2.
- Enabling OAM Rest OAP Calls
  In Oracle Access Manager 12c, WebGate interacts with Oracle Access Manager through REST API calls. In order for WebGate to have unrestricted access to these Rest APIs, you need to update the `*WEB_CONFIG_DIR*/webgate.conf` file.
- Adding a Load Balancer Certificate to WebGate
  Oracle WebGate 12c uses REST calls to interact with Oracle Access Manager 12c. To ensure that the communication works properly, you have to copy the load balancer certificates to WebGate Config and ensure that the REST endpoints are set correctly.
- Copying WebGates Artifacts to Web Tier
  When you created your Oracle Access Management installation, a WebGate called `Webgate_IDM` was created. In order for WebGate to communicate with the Access servers, you must copy the artifacts associated with this WebGate to the web tier.
- Restarting the Oracle HTTP Server Instance
- Setting Up the WebLogic Server Authentication Providers
  To set up the WebLogic Server authentication providers, back up the configuration files, set up the Oracle Access Manager Identity Assertion Provider and set the order of providers.
- Configuring Oracle ADF and OPSS Security with Oracle Access Manager
  Some Oracle Fusion Middleware management consoles use Oracle Application Development Framework (Oracle ADF) security, which can integrate with Oracle Access Manager Single Sign-on (SSO). These applications can take advantage of Oracle Platform Security Services (OPSS) SSO for user authentication, but you must first configure the domain-level `jps-config.xml` file to enable these capabilities.

About Oracle Webgate

Oracle WebGate is a web server plug-in that intercepts HTTP requests and forwards them to an existing Oracle Access Manager instance for authentication and authorization.

For Oracle Fusion Middleware 12*c*, the Oracle WebGate software is installed as part of the Oracle HTTP Server 12*c* software installation. See Registering and Managing OAM 11g

Agents in *Adminstrator’s Guide for Oracle Access Management*. Oracle WebGate is available for Oracle HTTP Server.

General Prerequisites for Configuring Oracle HTTP Server WebGate

Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.

For the most up-to-date information, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

For WebGate certification matrix, click and open http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/oam-webgates-2147084.html, then click the *Certification Matrix for 12c Access Management WebGates* link to download the certification matrix spreadsheet.

> **Note:**
>
> For production environments, it is highly recommended that you install Oracle Access Manager in its own environment and not on the machines that are hosting the enterprise deployment.

> **Note:**
>
> It is recommended that you use the WebGate version that is certified with your Oracle Access Manager deployment.

For more information about Oracle Access Manager, see the latest Oracle Identity and Access Management documentation, which you can find in the **Middleware** documentation on the Oracle Help Center.

Configuring Oracle HTTP Server 12*c* WebGate for an Enterprise Deployment

You need to perform the following steps in order to configure Oracle HTTP Server 12*c* WebGate for Oracle Access Manager on both WEBHOST1 and WEBHOST2.

In the following procedure, replace the directory variables, such as *WEB_ORACLE_HOME* and *WEB_CONFIG_DIR*, with the values, as defined in File System and Directory Variables Used in This Guide.

1. Perform a complete backup of the web tier domain.
2. Change directory to the following location in the Oracle HTTP Server Oracle home:

   ```
   cd WEB_ORACLE_HOME/webgate/ohs/tools/deployWebGate/
   ```

3. Run the following command to create the WebGate Instance directory and enable WebGate logging on OHS Instance:

   ```
   ./deployWebGateInstance.sh -w WEB_CONFIG_DIR -oh WEB_ORACLE_HOME
   ```

   For example:

   ```
   ./deployWebGateInstance.sh -w /u02/private/oracle/config/domains/ohsDomain/
   config/fmwconfig/components/OHS/ohs1 -oh /u01/oracle/products/web
   ```

4. Verify that a `webgate` directory and subdirectories was created by the `deployWebGateInstance` command:

   ```
   ls -lat WEB_CONFIG_DIR/webgate/
   total 16
   drwxr-x---+ 8 orcl oinstall 20 Oct  2 07:14 ..
   drwxr-xr-x+ 4 orcl oinstall  4 Oct  2 07:14 .
   drwxr-xr-x+ 3 orcl oinstall  3 Oct  2 07:14 tools
   drwxr-xr-x+ 3 orcl oinstall  4 Oct  2 07:14 config
   ```

5. Run the following command to ensure that the `LD_LIBRARY_PATH` environment variable contains `WEB_ORACLE_HOME/lib` directory path:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:WEB_ORACLE_HOME/lib
   ```

6. Change directory to the following directory

   ```
   WEB_ORACLE_HOME/webgate/ohs/tools/setup/InstallTools
   ```

7. Run the following command from the `InstallTools` directory.

   ```
   ./EditHttpConf -w WEB_CONFIG_DIR -oh WEB_ORACLE_HOME -o output_file_name
   ```

   For example:

   ```
   ./EditHttpConf -w /u02/private/oracle/config/domains/ohsDomain/config/
   fmwconfig/components/OHS/ohs1 -oh /u01/oracle/products/web
   ```

   > **Note:**
   >
   > The `-oh WEB_ORACLE_HOME` and `-o output_file_name` parameters are optional.

   This command:

   - Copies the `apache_webgate.template` file from the Oracle HTTP Server Oracle home to a new `webgate.conf` file in the Oracle HTTP Server configuration directory.
   - Updates the `httpd.conf` file to add one line, so it includes the `webgate.conf`.
   - Generates a WebGate configuration file. The default name of the file is `webgate.conf`, but you can use a custom name by using the `-o output_file_name` argument to the command.

Enabling OAM Rest OAP Calls

In Oracle Access Manager 12c, WebGate interacts with Oracle Access Manager through REST API calls. In order for WebGate to have unrestricted access to these Rest APIs, you need to update the *WEB_CONFIG_DIR*/webgate.conf file.

For example:

```
/u02/private/oracle/config/domains/webtier_domain/config/fmwconfig/
components/OHS/ohs1/webgate.conf
```

To update *WEB_CONFIG_DIR*/webgate.conf:

1. Add the following lines:

   ```
   <LocationMatch "/iam/access/binding/api/v10/oap">
       require all granted
   </LocationMatch>
   ```

2. Save the file.

Adding a Load Balancer Certificate to WebGate

Oracle WebGate 12c uses REST calls to interact with Oracle Access Manager 12c. To ensure that the communication works properly, you have to copy the load balancer certificates to WebGate Config and ensure that the REST endpoints are set correctly.

- Copying the LoadBalancer Certificates to WebGate Config
- Ensuring that the REST Endpoints are Set Correctly

Copying the LoadBalancer Certificates to WebGate Config

WebGate needs to trust your load balancer certificate. To ensure this trust, you should add the load balancer's certificate to the cacert.pem file, which is located in *WEB_CONFIG_DIR*/webgate/config.

You can obtain the certificate from the load balancer using a browser, such as Firefox. However, the easiest way to obtain the certificate is to use the openssl command. The syntax of the command is as follows:

```
openssl s_client -connect LOADBALANCER:PORT -showcerts </dev/null 2>/dev/null|
openssl x509 -outform PEM > LOADBALANCER.pem
```

For example:

```
openssl s_client -connect login.example.com:443 -showcerts </dev/null 2>/dev/
null|openssl x509 -outform PEM > login.example.com.pem
```

This command saves the certificate to a file named login.example.com.pem.

If you do not have load balancer certificates in your WebGate truststore, copy the `login.example.com.pem` file to *WEB_CONFIG_DIR*/webgate/config renaming it to `cacert.pem`.

For example:

```
cp login.example.com.pem WEB_CONFIG_DIR/webgate/config/cacert.pem
```

If you already have trusted certificates in WebGate, append the certificate to the `cacert.pem` file.

For example:

```
cp login.example.com.pem >> WEB_CONFIG_DIR/webgate/config/cacert.pem
```

Ensuring that the REST Endpoints are Set Correctly

To ensure that the REST endpoints are set correctly:

1. Log in to the OAM Console using your oam administrator account.
2. Click **Agents**.
3. Click **Search**.
4. Locate and click the name of the WebgGate agent from the search results to bring up the edit screen.
5. Expand the User Properties screen and check that the following parameters are defined correctly:
   - **OAMRestEndPointHostName** = `login.example.com`
   - **OAMRestEndPointPort** = `443`
   - **OAMServerCommunicationMode** = `HTTPS`

   These values should reflect the login point of your Oracle Access Manager installation. If unsure about these values, contact your OAM administrator.

Copying WebGates Artifacts to Web Tier

When you created your Oracle Access Management installation, a WebGate called `Webgate_IDM` was created. In order for WebGate to communicate with the Access servers, you must copy the artifacts associated with this WebGate to the web tier.

- Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location

Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location

After you run the idmTool configOAM, it creates a WebGate agent called Webgate_IDM. The process creates a number of artifacts relating to that agent in *IAD_ASERVER_HOME*/output/Webgate_IDM. These artifacts have to be copied to the Oracle HTTP Server configuration directory on the Web Tier hosts.

The location of the files in the Oracle HTTP Server configuration directory depends on the Oracle Access Manager security mode setting (OPEN, SIMPLE, or CERT).

The following table lists the required location of each generated artifact in the Oracle HTTP Server configuration directory, based on the security mode setting for Oracle Access Manager. In some cases, you might have to create the directories if they do not exist already. For example, the wallet directory might not exist in the configuration directory.

> **Note:**
>
> For an enterprise deployment, Oracle recommends simple mode, unless additional requirements exist to implement custom security certificates for the encryption of authentication and authorization traffic. The information about using open or certification mode is provided here as a convenience.
>
> Avoid using open mode, because in open mode, traffic to and from the Oracle Access Manager server is not encrypted.
>
> For more information about using certificate mode or about Oracle Access Manager supported security modes, see Securing Communication Between OAM Servers and WebGates in *Administrator's Guide for Oracle Access Management*.

Table 22-1 Web Tier Host Location to Copy the Generated Artifacts

File	Location When Using SIMPLE Mode	Location When Using CERT Mode
`wallet/cwallet.sso`[1]	*`WEB_CONFIG_DIR`*`/webgate/config/wallet/` By default the wallet folder is not available. Create the wallet folder under *`WEB_CONFIG_DIR`*`/webgate/config/`.	*`WEB_CONFIG_DIR`*`/webgate/config/wallet/`
`ObAccessClient.xml`	*`WEB_CONFIG_DIR`*`/webgate/config/`	*`WEB_CONFIG_DIR`*`/webgate/config/`
`password.xml`	*`WEB_CONFIG_DIR`*`/webgate/config/`	*`WEB_CONFIG_DIR`*`/webgate/config/`
`aaa_key.pem`	*`WEB_CONFIG_DIR`*`/webgate/config/simple/`	*`WEB_CONFIG_DIR`*`/webgate/config/`
`aaa_cert.pem`	*`WEB_CONFIG_DIR`*`/webgate/config/simple/`	*`WEB_CONFIG_DIR`*`/webgate/config/`

[1] Copy `cwallet.sso` from the wallet folder and not from the output folder. Even though there are 2 files with the same name they are different. The one in the wallet sub directory is the correct one.

> **Note:**
>
> If you need to redeploy the `ObAccessClient.xml` to `WEBHOST1` and `WEBHOST2`, delete the cached copy of `ObAccessClient.xml` and its lock file, `ObAccessClient.xml.lck` from the servers. The cache location on `WEBHOST1` is:
>
> ```
> WEB_DOMAIN_HOME/servers/ohs1/cache/
> ```
>
> And you must perform the similar step for the second Oracle HTTP Server instance on `WEBHOST2`:
>
> ```
> WEB_DOMAIN_HOME/servers/ohs2/cache/
> ```

Obtaining WebGate Artifacts

The easiest way to obtain the WebGate artifacts is to download them from the OAM console. To download, complete the following steps:

1. Log in to the OAM console with user 'oamadmin' using the following URL:

   ```
   http://IADADMINVHN.example.com:7001/oamconsole
   ```

2. Click **Agents**.
3. On the Search screen, click **Search**.
4. From the list of agents, select **Webgate_IDM** by clicking on its name.
5. Download the artifacts using the download button. A zip file gets downloaded on the host machine you are using.

Copy the downloaded zip file to the Oracle HTTP Server machine and unzip it to the *`WEB_CONFIG_DIR`*`/webgate/config` location. The files get extracted to the correct locations.

Restarting the Oracle HTTP Server Instance

For information about restarting the Oracle HTTP Server instance, see Restarting Oracle HTTP Server Instances by Using WLST in *Administering Oracle HTTP Server*.

If you have configured Oracle HTTP Server in a WebLogic Server domain, you can also use Oracle Fusion Middleware Control to restart the Oracle HTTP Server instances. See Restarting Oracle HTTP Server Instances by Using Fusion Middleware Control in *Administering Oracle HTTP Server*.

Setting Up the WebLogic Server Authentication Providers

To set up the WebLogic Server authentication providers, back up the configuration files, set up the Oracle Access Manager Identity Assertion Provider and set the order of providers.

The following topics assumes that you have already configured the LDAP authenticator by following the steps in Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group. If you have not already created the LDAP authenticator, then do so before you continue with this section.

> **Note:**
>
> You only need to perform these steps in the **IAMGovernanceDomain**, they will already have been performed in the **IAMAccessDomain** as part of running configOAM.

- Backing Up Configuration Files
- Setting Up the Oracle Access Manager Identity Assertion Provider
- Updating the Default Authenticator and Setting the Order of Providers

Backing Up Configuration Files

To be safe, you should first back up the relevant configuration files:

```
ASERVER_HOME/config/config.xml
ASERVER_HOME/config/fmwconfig/jps-config.xml
ASERVER_HOME/config/fmwconfig/system-jazn-data.xml
```

Also back up the `boot.properties` file for the Administration Server:

```
ASERVER_HOME/servers/AdminServer/security/boot.properties
```

Setting Up the Oracle Access Manager Identity Assertion Provider

Set up an Oracle Access Manager identity assertion provider in the Oracle WebLogic Server Administration Console.

To set up the Oracle Access Manager identity assertion provider:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. Click **Security Realms** in the left navigation bar.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. Click **New**, and select the asserter type **OAMIdentityAsserter** from the drop-down menu.
7. Name the asserter (for example, *OAM ID Asserter*) and click **OK**.
8. Click the newly added asserter to see the configuration screen for the Oracle Access Manager identity assertion provider.
9. Set the control flag to *REQUIRED*.
10. Under Chosen types, select both the **ObSSOCookie** and **OAM_REMOTE_USER** options, if they are not selected by default.
11. Click **Save** to save the settings.
12. Click **Activate Changes** to propagate the changes.

Updating the Default Authenticator and Setting the Order of Providers

Set the order of identity assertion and authentication providers in the WebLogic Server Administration console.

To update the default authenticator and set the order of the providers:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. From the left navigation, select **Security Realms**.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. From the table of providers, click the **DefaultAuthenticator**.
7. Set the Control Flag to `SUFFICIENT`.
8. Click **Save** to save the settings.
9. From the navigation breadcrumbs, click **Providers** to return to the list of providers.
10. Click **Reorder**.
11. Sort the providers to ensure that the OAM Identity Assertion provider is first and the DefaultAuthenticator provider is last.

Table 22-2 Sort order

Sort Order	Provider	Control Flag
1	OAMIdentityAsserter	`REQUIRED`
2	LDAP Authentication Provider	`SUFFICIENT`
3	OIMAuthenticationProvider	`SUFFICIENT`
4	DefaultIdentityAsserter	`N/A`
5	Trust Service Identity Asserter	`N/A`
6	DefaultAuthenticator	`SUFFICIENT`

12. Click **OK**.
13. Click **Activate Changes** to propagate the changes.
14. Shut down the Administration Server, Managed Servers, and any system components, as applicable.
15. Restart the Administration Server.
16. If you are going to configure ADF consoles with SSO, you can keep the managed servers down and restart them later. If not, you need to restart managed servers now.

Configuring Oracle ADF and OPSS Security with Oracle Access Manager

Some Oracle Fusion Middleware management consoles use Oracle Application Development Framework (Oracle ADF) security, which can integrate with Oracle Access Manager Single Sign-on (SSO). These applications can take advantage of Oracle Platform Security Services

(OPSS) SSO for user authentication, but you must first configure the domain-level `jps-config.xml` file to enable these capabilities.

The domain-level `jps-config.xml` file is located in the following location after you create an Oracle Fusion Middleware domain:

```
ASERVER_HOME/config/fmwconfig/jps-config.xml
```

> **Note:**
>
> The domain-level `jps-config.xml` should not be confused with the `jps-config.xml` that is deployed with custom applications.

To update the OPSS configuration to delegate SSO actions in Oracle Access Manager, complete the following steps:

1. Change to the following directory:

   ```
   ORACLE_COMMON_HOME/common/bin
   ```

2. Start the WebLogic Server Scripting Tool (WLST):

   ```
   ./wlst.sh
   ```

3. Connect to the Administration Server, by using the following WLST command:

   ```
   connect('admin_user','admin_password','admin_url')
   ```

4. Run the `addOAMSSOProvider` command, as shown:

   ```
   addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication",
   logouturi="/oam/logout.html")
   ```

   The following table defines the expected value for each argument in the `addOAMProvider` command.

   > **Note:**
   >
   > Perform this action for each domain in your configuration.

Table 22-3 Expected Values for the Argument in the `addOAMProvider` command

Argument	Definition
loginuri	Specifies the URI of the login page **Note:** For ADF security enabled applications, "/*context-root*/adfAuthentication" should be provided for the 'loginuri' parameter. For example: `/${app.context}/adfAuthentication` **Note:** `${app.context}` must be entered as shown. At runtime, the application replaces the variable appropriately. Here is the flow: **a.** User accesses a resource that has been protected by authorization policies in OPSS, fox example. **b.** If the user is not yet authenticated, ADF redirects the user to the URI configured in *loginuri*. **c.** Access Manager, should have a policy to protect the value in *loginuri*: for example, "/*context-root*/ adfAuthentication". **d.** When ADF redirects to this URI, Access Manager displays a Login Page (depending on the authentication scheme configured in Access Manager for this URI).
logouturi	Specifies the URI of the logout page. The value of the *loginurl* is usually `/oam/logout.html`.
autologinuri	Specifies the URI of the autologin page. This is an optional parameter.

5. Disconnect from the Administration Server by entering the following command:

   ```
   disconnect()
   ```

6. Restart the Administration Server and the managed servers.

23

Sanity Checks

The sanity tests described in this chapter are over and above the normal tests detailed in the guide. They are designed to test the in-depth functionality of Oracle Access Management (OAM) and Oracle Identity Manager (OIM).
This chapter includes the following topics:

Sanity Checks for Oracle Access Management

Learn about the sanity checks applicable for Oracle Access Management (OAM).

Verifying LDAP Authentication for OAM Agent Protected Application for Valid User

To verify the LDAP authentication for OAM agent protected application for valid user, do the following:

1. Access an application protected by an OAM WebGate which is configured to OAM server.
2. Check out the URL that is being redirected to for authentication is from OAM server.
3. Provide a valid username and password from the OUD authentication form and click Login.
4. Check the cookies that are created in the browser.

Expected Result:

- OAM agent protected Application can be accessed on providing valid credentials.
- ObSSOcookie and OAM_ID cookies are created in the browser session.

Verifying LDAP Authentication Failure for OAM Agent Protected Application for Invalid Password

To verify the LDAP authentication failure for OAM agent protected application for invalid password, do the following:

1. Access an application protected by an OAM WebGate which is configured to OAM server.
2. Check out the URL that is being redirected to for authentication is from OAM server.
3. Provide a valid username and an invalid password in the authentication form.

Expected Result:

- User authentication fails.
- Appropriate error message is displayed.
- Resource cannot be accessed by the user.

Verifying LDAP Authentication Failure for OAM Agent Protected Application for Invalid Username

To verify the LDAP authentication failure for OAM agent protected application for invalid username, do the following:

1. Access an application protected by an OAM WebGate which is configured to OAM server.
2. Check out the URL that is being redirected to for authentication is from OAM server.
3. Provide an invalid username and any password in the authentication form.

Expected Result:

- User authentication fails.
- Appropriate error message is displayed.
- Resource cannot be accessed by the user.

Verifying Access of OAM Agent Protected Unavailable Resource

If you access an OAM agent protected unavailable resource, an appropriate error message is displayed though the credentials provided are valid. To verify this, do the following:

1. Access a resource url protected by an OAM WebGate which is configured to OAM server when that resources is not available.
2. Check out the URL that is being redirected to for authentication is from OAM server.
3. Provide a valid username and password in the authentication form.
4. Check the cookies that are created in the browser.

Expected Result:

OAM WebGate protected application cannot be accessed and a proper error message should be displayed.

Verifying Access of Resource that was Recently Deleted or Replaced from the Policy

If you access a resource which was recently deleted or replaced from the policy, the authentication is not required and the access is granted. To verify this, do the following:

1. Remove a resource and replace it with new one in the `policy.xml` or UI.
2. Access the application or resource that you deleted or replaced in the previous step. This application must be protected by an OAM WebGate which is configured to OAM server.

3. Check if the user is not asked for authentication without having to restart the OAM 11*g* Server or WebLogic Server.
4. Check if user is able to access the resource.

Expected Result:

Resource or Application can be accessed without having to authenticate user and without having to restart the OAM 11*g* Server or WebLogic Server.

Sanity Checks for Oracle Identity Governance

Learn about the sanity checks applicable for Oracle Identity Governance (OIG).

- Creating Organization
- Creating a User Name
- Creating Role
- Managing Sandboxes
- Publishing a Sandbox
- Adding User Defined Field (UDF) for a User
- Creating a Disconnected Application and Provision
- Importing and Configuring DB User Management
- Creating an Access Policy and Provision
- Creating End User Request for Accounts, Entitlements, and Roles
- Resetting Account Password
- Creating a Certification and Approving
- Creating Identity Audit Scan Definitions and Viewing its Results
- Testing Identity Audit

Creating Organization

To create an organization, do the following:

1. Log in to the Identity Console as `xelsysadm` using the following URL:

   `https://`prov.example.com`/identity`
2. Click **Manage**, and then click **Organization**.
3. Click **Create**, and specify the org name as TestOrg.
4. After you have entered the details of your organization, click **Save** to store the changes.

Creating a User Name

To create a user, do the following:

1. Log in to the Identity Console as `xelsysadm` using the following URL:

   `https://`*`prov.example.com`*`/identity`
2. Click **Manage**, and then click **User**.
3. Click **Create**, and specify the user name as Rahul Dravid.

4. Select Org as **TestOrg**.
5. Set and confirm user password.
6. Log in as Rahul Dravid.
7. Set the challenge questions and answers.
8. Log in to the Identity Console and verify the user name.

Creating Role

To create a role, do the following:

1. Log in to the Identity Console as `xelsysadm` using the following URL:

   `https://prov.example.com/identity`
2. Click **Manage**, and then click **Roles and Access Policies > Roles**.
3. Click **Create** and provide the mandatory attributes (Name, Display Name) to create a Role named `Coach`.
4. Click **Next** repeatedly until the Publish Role to Organizations page is displayed.
5. On the **Organizations** page, click **Add Organizations**. Provide the organization name as **TestOrg** and click **Search**.
6. Select the organization **TestOrg** and click **Add Selected**. Click **Select**.
7. Click **Next**, and then click **Finish**.

Managing Sandboxes

A number of the operations below require the creation of a sandbox. A sandbox is a non-active area where things can be tried out prior to making them live.

Creating a Sandbox

You can crate sandboxes either from the System Administration Console or the Identity Console. The steps are the same. The following is an example for creating a sandbox in the System Administration Console.

To create a sandbox:

1. Log in to the System Administration Console as `xelsysadm` using the following URL:

   `http://IGDADMIN.example.com/sysadmin`
2. Click **Sandboxes**.
3. Click **Create Sandbox**.
4. Enter the below details in the Create Sandbox window.

   Table 23-1 Properties of the Sandbox Window

Attribute	Value
Name	TestSandbox
Description	Enter a description

   Select **Activate Sandbox**.

5. Click **Save and Close**.

Publishing a Sandbox

Once the changes are fine, you publish the sandbox to make it live. This is achieved by performing the following steps:

1. Log in to the System Administration Console as `xelsysadm` using the following URL:

   `http://IGDADMIN.example.com/sysadmin`

2. Click **Sandboxes**.
3. A window appears with a list of the sandboxes.
4. Click a sandbox. For example: **Test Sandbox**.
5. Click **Publish Sandbox** to make the changes active.

Adding User Defined Field (UDF) for a User

To add a User Defined Field (UDF):

1. Log in to the System Administration Console as `xelsysadmin` using the following URL:

   `http://IGDADMIN.example.com/sysadmin`

2. Create & Activate Sandbox.
3. Click **User** from under **System Entities**.
4. Click **Create** under **Action**.
5. Select **Text** and click **OK**.
6. Enter **Display Label** and **Name**, select **Searchable**, and click on **Save and Close**. You have now created a user defined field (UDF) with the name you specified.
7. Publish Sandbox.
8. Log in to the Identity Console as `xelsysadm` using the following URL:

   `http://prov.example.com/identity`

9. Create and Activate Sandbox.
10. Click on **Manage** to show the management menu.
11. Open Users page, and click **Create**.
12. Click **Customize** at the top right of the screen.
13. Enter the details for all the attributes listed below.

Table 23-2 User Defined Field Properties

Attribute	Description
First Name	Enter a name for example: John
Last Name	Enter a last name for example: Doe
Email	Enter an email address for example: john.doe@example.com
Organisation	Enter or search for an Organization for example: TestOrg

Table 23-2 (Cont.) User Defined Field Properties

Attribute	Description
User Type	Select the type of user from the drop down list.
User login	Enter the users login name for example: JohnDoe
Password	Enter an initial password for the user to use.

14. Go to the **Structure** tab at the top left of the screen.
15. The user entry screen is displayed. Scroll down until you see the Basic Information section. As you move down the screen, certain areas are highlighted by a box. When the Entire Basic Information section is highlighted, including the title, click it. A dialogue box is displayed confirming that you want to edit the task flow. Click **Edit**. A structure window is displayed on the right.
16. Click **PanelForm Layout**.
17. Click **Add Content**.
18. Select **Data Component - Catalog**, and then click **UserVO**.
19. Find the User Defined Field you created in step 6 and Click **Add**. Select ADF Input Text w/ Label. Your user defined Field will now be shown in the Basic Information section of the User screen.
20. Close the Add Content Selection screen.
21. Click **Close** at the top of the screen to close the editing window.
22. Close the structure form by clicking **Close** on the top right corner of the Identity Console window.
23. Publish the sandbox.
24. Log out and log in again.
25. Open the User Details page.
26. Create a user name populating the user defined field that you created in step 6 and verify if it is displayed properly in the user details page.

Creating a Disconnected Application and Provision

To create a disconnected application and provision:

1. Create a lookup by completing the following steps:
    a. Log in to the System Administration Console as `xelsysadm` using the following URL:

       `http://`*`igdadmin.example.com`*`/sysadmin`

    b. Go to **System Configuration** tab and click **Lookups**.
    c. Click the **Create** link under Action drop down list.
    d. Enter the meaning as `Lookup.Disc`, and enter the code as `Lookup.Disc`.
    e. Click **Create link** under **Action** drop down list.
    f. Enter the value `HDD` for Meaning, and `HDD` for Code.
    g. Repeat with the values of `CD` and `CD` for Meaning and Code.
    h. Click **Save**.

i. Enter the value `Lookup.Disc` for Meaning, `Lookup.Disc` for code, and click **Search**.

j. The values **HDD** and **CD** are displayed. Click **OK**.

2. Create disconnected application instances by completing the following steps:

a. Log in to the System Administration Console as `xelsysadm` using the following URL:

   `http://`*`igadmin.example.com`*`/sysadmin`

b. Click the **Sandboxes** link, and then click **Create Sandbox**.

c. Enter the name **Disc**, and click **Save** and **Close**. Click **OK** to confirm. Sandbox is activated.

d. Go to **Provisioning configuration**, and click **Application Instances**.

e. Click **Create**. The Create App Instance page is displayed by enabling the Attribute tab.

f. Enter the name as **Disc**, Description as **Disc**, and check the **Disconnected** check box. Click **Save**. Click **OK** to confirm. Feedback message is displayed to confirm that Application Instance Disc is created successfully.

g. On the same page, go to the **Attribute** tab. Form field is added with the name **Disc**. Click **Edit** next to Form field.

   This step enables the Manage Disc tab with its subtab, **Fields**, opened. Click the **Child Objects** tab which is next to the **Fields** tab.

h. Click **Add**, and enter the name as **chdisc**, description as **chdisc**, and Click **OK**.

i. Click **chdisc**. This opens another page by enabling the **Fields** tab.

j. Click **Create link** under **Action** drop down list and select **Lookup** as the Field type, and click **OK**.

k. Enter Display Label and name as `Disc`, select **Searchable**. Click **Lookup Type**, and then click **Search** or look up icon (Magnifier icon). Enter the meaning as `Lookup.Disc`.

l. Click **Search**. Values **HDD** and **CD** must be displayed. Click **OK**. Lookup must be selected. Default Value Label, One Drop down gets added. Click on that, and you will see the values: HDD and CD.

   If you enabled **Entitlement**, make sure that **Searchable** and **Searchable Picklist** are also selected. Keep the remaining ones with the default values.

m. Click **Save and Close**.

n. Click **Back to Parent Object**, and then click **Regenerate view**.

o. Enable **Parent Form + Child Tables (Master/Detail)**, keep the default setting. Click **OK**.

p. Go to the **Application Instance** tab. Search for an Application Instance **Disc**.

q. Click **Refresh**, and click **Apply** on Disc form.

r. Go to **System Configuration** > **Scheduler** from the left navigation window.

s. Enter the value **Ent*** in the **Search Scheduled Jobs** field, click **Search** or **Go** button.

t. The results are displayed. Click on Entitlement List job name.

u. Click **Run now**. A confirmation message is displayed saying the Job is running.

v. Click **Refresh.** Verify that the execution status is successful. Close the window.

w. Go to the Application instance's Entitlement tab. Two entitlements are displayed - HDD, CD. Select either of the the two and click **Assign +** from the window below.

x. Search organization name, by entering the value Top, and click **Search**.

y. Top organization should be displayed. Select that row / organization, and click **Add Selected**. Selected organization gets added successfully.

z. Check **Apply to Entitlement**, and click **Select**. Selected Organization gets added successfully.

aa. Click **Assign**.

ab. Repeat steps x, y, z, and aa for the CD row.

ac. Search for the organization name **TestOrg**, and click **Search**.

ad. **TestOrg** organization is displayed. Select that row / organization, and click **Add Selected**.

ae. Selected organization gets added successfully. Check **Apply to Entitlement** and click **Select**. Selected organization gets added successfully.

af. Go to the Application Instance's Attribute tab. Click **Apply**. A message is displayed stating that the Application instances disc is modified successfully.

ag. Click Sandboxes.

ah. Select the same sandbox **Disc**. Click **Export sandbox** button. Export sandbox generate .zip file `sandbox_disc.zip`. Click **OK** button. Zip file is saved and generated.

ai. After export is successfully completed, click **Publish sandbox** button. Click **Yes** to confirm.

aj. After you publish, the sandbox is listed under Publish Sandboxes link.

3. Provision the disconnected application instances and entitlements to user by completing the following steps:

   a. Log in to the Identity Console as `xelsysadm` using the following URL:

      https://*prov.example.com*/identity

   b. Click **Manage** and then click **Users**.

   c. Search for the user name `Rahul Dravid`, and click **Search**.

   d. The user Rahul Dravid is displayed. Click on that user link. User details are displayed.

   e. Go to **Accounts** tab, and then to the **Request Account** tab. Account access request page is displayed. Select **Enabled Add access**., and go to the **Catalog** tab. All available Application Instances are displayed.

   f. Click **Add to cart** of the **Disc** Disconnected application instances, and click **Next**. The cart detail page is displayed

   g. Click the Pen icon in the Request detail pane.

   h. Enter the account logging name as `Rahul Dravid_123`, and the password as <*password*>. Click **Update**.

   i. Click **Submit**. Request will be generated with a message `Request for access completed successfully`.

   j. Go to the **Self Service** tab. Click **Provisioning task**, and the go to the **Manual Fulfillment** tab. Manual fulfillment page is displayed.

   k. Click on that request. Request details are displayed. Verify the data. Click **Complete**, and then click **Refresh**.

   l. Go to the **Manage** tab, and then to the **User** tab. Open the same user Rahul Dravid.

m. Go to the **Account** tab. Click **Refresh**. Verify that the account status is `Provisioned`.

n. Select the same account name Rahul Dravid_123, and click Request Entitlement button. Entitlement Access request page is displayed. Enable **Add Access** and go to the **Catalog** tab.

o. Click **Add to cart** for entitlement `HDD`. Click **Next**.

p. Click **Submit**. Request will be generated with a message "Request for access completed successfully".

q. Go to the **Self service** tab. Click on **Provisioning task**, and go to **Manual Fulfillment** tab. Manual fulfillment page is displayed

r. Click on that request. Request details are displayed. Verify the data. Click **Complete**, and then click **Refresh**.

s. Go to the **Manage** tab, and then to the **User** tab. Open the details of the same user - Rahul Dravid.

t. Go to the **Entitlement** tab. Click **Refresh**. Verify that the Entitlement status is **Provisioned**.

Importing and Configuring DB User Management

To import and configure database user management:

1. Download the latest Database User Management Connector from the Oracle Identity Manager Connector Downloads page on Oracle Technology Network (OTN):

   http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html

2. Log in to the System Administration Console as `xelsysadmin` user using the following URL:

   `http://`*`igdadmin.example.com`*`/sysadmin`

3. Go to the **System Configuration** tab and click **Import**.

4. Select the file `DBUserManagement-Oracle-ConnectorConfig.xml`'. Sample location: `D:\DBUM-12.2.1.4.0\xml`

5. Click **Open**.

6. Click **Next**. You can either provide the ITResource details now or later. To provide the same later, click **Next**.

7. Click **Selected Entities** to view selections, and click **Import**. Once the import is successfully completed, click **OK**.

8. Copy the third party jars of target systems to the *`IGD_ORACLE_HOME`*`/idm/server/ConnectorDefaultDirectory/targetsystems-lib/DBUM-12.2.1.4.0` directory.

   > **Note:**
   >
   > If the target is Oracle database, no driver jar is needed.

9. To configure a trusted source reconciliation, create and configure a new IT resource. For example, Oracle DB Trusted of type Oracle DBUM.

10. In the Configuration Lookup, update the trusted configuration lookup name as `Lookup.DBUM.Oracle.Configuration.Trusted`. This configures the ITResource for the target system.

11. Either you can create the ITResource and provide the following details or Open the existing ITResource 'Oracle DB' as specified below:

    ITResource Details:

    `Configuration Lookup = Lookup.DBUM.Oracle.Configuration`

    `Connector Server Name =`

    `Connection Properties =` Specify the connection properties for the target system database.

    `Database Name =` This field identifies database type (such as Oracle and MSSQL) and its used for loading respective scripts. Sample value: Oracle

    `JDBC Driver = oracle.jdbc.driver.OracleDriver`

    `JDBC URL =` For Oracle: `jdbc:oracle:thin:@host:port:sid`

    `Login Password =` Enter the password for the user name of the target system account to be used for connector operations.

    `Login User = sys as sysdba`

Creating an Access Policy and Provision

To create an access policy and provision:

1. Log in to the Identity Console as `xelsysadm` using the following URL: https://prov.example.com/identity.
2. Click **Manage**.
3. Click **Roles and Access Policies** -> **Roles**.
4. Create a Role named `DBUMRole`.
5. Click **Home** tab to select the main management options.
6. Click **Users**.
7. Click **Create**.
8. Create an user named `Jean Wilson`.
9. Click **Home** tab.
10. Click **Roles and Access Policies** -> **Roles**.
11. Select the Role `DBUMRole`.
12. The role page is displayed - Click **Members**.
13. Click **Add**.
14. In the add members dialogue box, search for the user Jean Wilson.
15. Click the user Jean Wilson.
16. Click **Add Selected**.
17. Click **Apply**.
18. Create another user named `Patrick Morgan` and assign the user role `DBUMRole`.
19. Click **Manage** and click **Home**tab.
20. Open the user details page of Jean Wilson and click **Accounts** tab. DBUM Account should be in Provisioned state.

21. Go to the **Entitlements** tab and verify all child data added are displayed.
22. Repeat the previous two steps for user Patrick Morgan.

Creating End User Request for Accounts, Entitlements, and Roles

To create an end user request for roles, do the following:

1. Create a user `Arthur Hill`.
2. Log in as `Arthur Hill` and open **My Access** page, and then **Roles**.
3. Click **Request** and in catalog, add **DBUMRole** to cart.
4. Submit request.
5. Log in as administrator and open Pending Approvals.
6. Open the request and approve.
7. As `Arthur Hill`,verify that the role is assigned successfully.

To create an end user request for accounts, do the following:

1. Create a user `Bruce Parker`.
2. Log in as `Bruce Parker` and open **My Access** page, and then **Roles**.
3. Click **Request**.
4. From the Catalog, select the **DBUM App** and add to cart.
5. Click **Next** and click **Submit** to submit the request.
6. Log in as administrator and open Inbox.
7. Open the request, verify the details, and approve request.
8. As `Bruce Parker`, verify that the Account is provisioned successfully.

To create an end user request for entitlements, do the following:

1. Log in as `Jean Wilson`.
2. Open the **My Access** page and go to the **Accounts** tab.
3. Select the **DBUM app**, and click **Request Entitlements** under Action.
4. Add any entitlement to cart and submit request.
5. Log in as administrator and open Inbox.
6. Open the request and approve.
7. As Jean Wilson, verify that the entitlement is provisioned successfully.

Resetting Account Password

To reset the account password:

1. Log in to the Identity Console as `Jean Wilson`.
2. Click **My Access** and go to the **Accounts** tab.
3. Select **SSOTarget** and click **Reset Password** in Action.
4. Provide a new password and submit.
5. Log out and re-login as `xelsysadm`.

6. Click **Manage** and then click **Users**.
7. Search for Jean Wilson and open the user details page.
8. Go to the **Accounts** tab and select **DBUM App**.
9. Click **Resource History** under Action and check if the Password Updated task is triggered and is in Completed status.

Creating a Certification and Approving

Complete the following prerequisites to create a certification and approve:

1. Log in to Identity Console as `xelsysadm`.
2. Launch the System Administration Console.
3. Go to the **System Configuration** tab and click **Configuration Properties**.
4. Look for the following system properties:

   ```
   Property name = Identity Auditor Feature Set Availability
   ```

   ```
   Keyword = OIG.IsIdentityAuditorEnabled
   ```

   ```
   Value = TRUE
   ```

5. Save the setting.
6. Restart the OIM server to see the Compliance tab in Identity Console.

To create a certification and approve:

1. Log in to the Identity Console as `xelsysadm`.
2. Go to **compliance**, **Identity Certification**, and then **Definitions**.
3. Create a user type certification with the following information:
   - General details page: Enter the name = `UserCertification`, Type = `user`; Enter Description and click **Next**.
   - Base Selection page: Selected only **Users from Selected Organization** and Add organization (TestOrg). Added organization is displayed. Select **Users with Any Level of Risk** as Risk Level, and click **Next**.
   - Content selection page: Keep the default values, and click **Next**.
   - Configuration page: Keep the default and click **Next**.
   - Select the reviewer by searching for a user, for example, MSDhoni, and click **Next**.
   - Disable Incremental, and click **Next**.
   - Summary page: Click **Create**, and click **Yes** to confirm. Certification is created successfully.
4. Log in to the System Administration Console as `xelsysadm`.
5. Click **Scheduler**.
6. Search for a certification `cert_UserCertification`. Verify that the job is run successfully.
7. Log in to the Identity Console as `xelsysadm`, and log out.
8. Log in to the Identity Console as a reviewer (MSDhoni).
9. Go to **Self service**, and click **Certification**.
10. Open the same certification **UserCertification [MSDhoni]**.

11. Certification details are displayed. You will see the user "Rahul Dravid".
12. Select user Rahul Dravid.
13. Verify, Role - Coach, Account - Disc, Entitlement - HDD.
14. Select all rows, and take the Complete action. Sign-off pop up should be displayed.
15. Enter the password (username = MSDhoni ; Password = <*password*>). Click **OK**. Certification is completed successfully. It should now reflect in your Inbox.
16. Log in to the Identity Console as MSDhoni / Xelsysadm.
17. Go to **Complaince**, **Identity Certification**, and then **Dashboard**. Dashboard details are displayed.
18. Select **Completed** from the Show Label. This displays all of the completed certifications.

Creating Identity Audit Scan Definitions and Viewing its Results

Complete the following prerequisites to create identity audit scan definitions:

1. Log in to the Identity Console as `xelsysadm`.
2. Launch the System Administration Console.
3. Go to the **System Configuration** tab, and click **Configuration Properties**.
4. Look for the following system properties:

   `Property name = Identity Auditor Feature Set Availability`

   `Keyword = OIG.IsIdentityAuditorEnabled`

   `Value = TRUE`
5. Save the setting.
6. Restart the OIM server to See the Compliance tab in the Identity Console.

To create a rule:

1. Log in to the Identity Console as `xelsysadm`.
2. Click **Compliance**, and then click **Identity Audit**.
3. Select **Rules**, and click **Create**.
4. Create an identity rule `Identity Rule 1` by the following condition builder:

   `user.Display Name; Equals ; Rahul Dravid`
5. Click **Create**. The rule is created.

To create a policy:

1. Log in to the Identity Console as `xelsysadm`.
2. Click **Compliance** and then click **Identity Audit**.
3. Click **Policies**, and click **Create**.
4. Create a policy `Identity Policy 1` by adding the rule `Identity Rule 1`.
5. Click **Create**.

To create scan definition:

1. Log in to the Identity Console as `xelsysadm` using the following URL:

   `https://`*`prov.example.com`*`/identity`

2. Click **Compliance** and then click **Identity Audit**.
3. Click **Scan definitions**, and then click **Create**.
4. Create a scan definition `Identity Scan 1` by adding the policy `Identity Policy 1`.
5. On the Base selection page, select all users.
6. On the Configuration page, keep the default values.
7. On the Summary page, click **Finish**. Scan definition is added successfully.
8. Run the scan definition by selecting **Identity Scan 1**, and clicking **Run now**. Verify that the scan definition is run successfully.
9. Preview the scan definition result by doing the following:
   a. After you run the scan definition, select the scan definition row or record **Identity Scan 1**.
   b. Click **View Scan**. The scan definition results are displayed.

Testing Identity Audit

Complete the following steps to enable audit feature in Oracle Identity Manager:

1. Log in to the System Administration console.
2. Click **System Properties** under **System Configuration**.
3. Search for the property `OIG.IsIdentityAuditorEnabled` and update the property value to `TRUE`.
4. Restart the Oracle Identity Manager Managed Server for the change to take effect.
5. Log in to the Identity console as `xelsysadm` using the following URL:

   `https://`*`prov.example.com`*`/identity`
6. Click **Compliance** and then click **Reports**.

   Verify that the Reports page is opened successfully.

24

Troubleshooting

You can troubleshoot the common issues that may arise with the Identity and Access Management enterprise deployment. The solutions provided for the common problems help you resolve them quickly.

This chapter includes the following topics:

- Troubleshooting IDMLCM Start/Stop Scripts
  Learn about the issue related to starting or stopping the Managed server using the Start/Stop scripts and the solution to fix the issue.
- Troubleshooting Oracle Access Management Access Manager
  Learn about some of the common problems that you may encounter with Oracle Access Manager and the actions you can take to resolve them.
- Troubleshooting Oracle Identity Governance
  Learn about some of the common problems that may arise with Oracle Identity Manager and the actions you can take to resolve the problem.
- Troubleshooting Oracle SOA Suite
  Learn about the transaction timeout error that may arise with Oracle SOA Suite and the action you can take to resolve the problem.
- Troubleshooting Integration OIGOAMIntegration.sh-configureLDAPConnector
  Learn about the error you may encounter during the inegration process and the solution to fix this error.
- General Troubleshooting
  Learn about the error you may encounter when starting the Managed Server from the WebLogic Console and the resolution to fix the error.

Troubleshooting IDMLCM Start/Stop Scripts

Learn about the issue related to starting or stopping the Managed server using the Start/Stop scripts and the solution to fix the issue.

- Start/Stop Scripts Fail to Start or Stop a Managed Server

Start/Stop Scripts Fail to Start or Stop a Managed Server

Problem

Problem: Start/Stop scripts fail to start or stop a managed server.

The start/stop logs in the directory *SHARED_CONFIG_DIR*/scripts/logs contain an error similar to this:

```
weblogic.utils.AssertionError: ***** ASSERTION FAILED *****
        at
weblogic.server.ServerLifeCycleRuntime.getStateRemote(ServerLifeCycleRuntime.java:734)
        at
weblogic.server.ServerLifeCycleRuntime.getState(ServerLifeCycleRuntime.java:581)
```

```
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
```

Solution

1. Shut down the failing managed server. You might have to kill the process.
2. Back up the managed server's LDAP data, then remove it. For example:

   ```
   rm -rf PRIVATE_CONFIG_DIR/domains/IAMAccessDomain/servers/server_name/data/ldap
   ```

   where *server_name* is the name of the failing managed server.
3. Restart the managed server.

Troubleshooting Oracle Access Management Access Manager

Learn about some of the common problems that you may encounter with Oracle Access Manager and the actions you can take to resolve them.

- Access Manager Runs out of Memory
- User Reaches the Maximum Allowed Number of Sessions
- Policies Do Not Get Created When Oracle Access Management Access Manager is First Installed
- You Are Not Prompted for Credentials After Accessing a Protected Resource
- Cannot Log In to Access Management Console
- Oracle Coherence Cluster Startup Errors in WLS_AMA Server Logs
- Errors in log File when Starting OAM Servers
- Too Many Redirects Error in Browser

Access Manager Runs out of Memory

Problem

After Access Manager has been running for a while, you see the following error message in the output:

```
Attempting to allocate 1G bytes
There is insufficient native memory for the Java Runtime Environment to continue.
```

Possible reasons

- The system is out of physical RAM or swap space.
- In 32 bit mode, the process size limit was reached.

Solutions

- Reduce memory load on the system.
- Increase physical memory or swap space.
- Check if swap backing store is full.
- Use 64 bit Java on a 64 bit OS.
- Decrease Java heap size (-Xmx/-Xms).
- Decrease number of Java threads.

- Decrease Java thread stack sizes (-Xss).
- Disable compressed references (-XXcompressedRefs=false).
- Ensure that command line tool `adrci` can be executed from the command line.
  - at oracle.dfw.impl.incident.ADRHelper.invoke(ADRHelper.java:1309)
  - at oracle.dfw.impl.incident.ADRHelper.createIncident(ADRHelper.java:929
  - at oracle.dfw.impl.incident.DiagnosticsDataExtractorImpl.createADRIncident(DiagnosticsDataExtractorImpl.java:1116)
- On both OAMHOST1 and OAMHOST2, edit the file `setSOADomainEnv.sh`, which is located in *`IAD_MSERVER_HOME`*`/bin` and locate the line which begins:

  ```
  PORT_MEM_ARGS=
  ```

  Change this line so that it reads:

  ```
  PORT_MEM_ARGS="-Xms768m -Xmx2560m"
  ```

User Reaches the Maximum Allowed Number of Sessions

Problem

The Access Manager server displays an error message similar to this:

```
The user has already reached the maximum allowed number of sessions. Please
close one of the existing sessions before trying to login again.
```

Solution

If users log in multiple times without logging out, they might overshoot the maximum number of configured sessions. You can modify the maximum number of configured sessions by using the Access Management Administration Console.

To modify the configuration by using the Access Management Administration Console, proceed as follows:

1. Go to **System Configuration** -> **Common Settings** -> **Session**
2. Increase the value in the **Maximum Number of Sessions per User** field to cover all concurrent login sessions expected for any user. The range of values for this field is from 1 to any number.

Policies Do Not Get Created When Oracle Access Management Access Manager is First Installed

Problem

The Administration Server takes a long time to start after configuring Access Manager.

Solution

Tune the Access Manager database. When the Administration Server first starts after configuring Access Manager, it creates a number of default policies in the database. If the database is distant or in need of tuning, this can take a significant amount of time.

```
Resources
Authentication Policies
   Protected Higher Level Policy
   Protected Lower Level Policy
   Publicl Policy
Authorization Policies
   Authorization Policies
```

If you do not see these items, the initial population has failed. Check the Administration Server log file for details.

You Are Not Prompted for Credentials After Accessing a Protected Resource

Problem

When you access a protected resource, Access Manager should prompt you for your user name and password. For example, after creating a simple HTML page and adding it as a resource, you should see credential entry screen.

Solution

If you do not see the Credential Entry screen, perform the following steps:

1. Verify that host aliases for IAMAccessDomain have been set. You should have aliases for `IAMAccessDomain`:80, `IAMAccessDomain`:Null, `IADADMIN.example.com:80`, and `login.example.com:443`, where Port `80` is *`HTTP_PORT`* and Port `443` is *`HTTP_SSL_PORT`*.
2. Verify that WebGate is installed.
3. Verify that `ObAccessClient.xml` was copied from *`IAD_ASERVER_HOME`*`/output` to the WebGate Lib directory and that OHS was restarted.
4. When you first created the `ObAccessClient.xml` file, it was not formatted. When you restart OHS, re-examine the file to ensure that it is formatted. OHS gets a new version of the file from Access Manager when it first starts.
5. Shut down the Access Manager servers and access the protected resource. If you do not see an error saying Access Manager servers are not available, re-install WebGate.

Cannot Log In to Access Management Console

Problem

You cannot log in to the Access Management Console. The Administration Server diagnostic log might contain an error message similar to this:

```
Caused by: oracle.security.idm.OperationFailureException:
oracle.security.am.common.jndi.ldap.PoolingException [Root exception is
oracle.ucp.UniversalConnectionPoolException:
Invalid life cycle state.
 Check the status of the Universal Connection Pool]
        at
oracle.security.idm.providers.stdldap.UCPool.acquireConnection(UCPool.java:112)
```

Solution

Remove the `/tmp/UCP*` files and restart the Administration Server.

Oracle Coherence Cluster Startup Errors in WLS_AMA Server Logs

Problem

The WLS_AMA2 server has oam application deployment in failed state. The WLS_AMA2 server logs report request timeout exceptions while starting the cluster service, similar to following logs:

```
Oracle Coherence GE 3.7.1.13 <Warning> (thread=Cluster, member=n/a): Delaying
formation of a new cluster; IpMonitor failed to verify the reachability of senior
Member(Id=1, Timestamp=, Address=, MachineId=,
Location=site:,machine:IADADMINVHN,process:8499, Role=WeblogicServer); if this
persists it is likely the result of a local or remote firewall rule blocking
either ICMP pings, or connections to TCP port 7>

Error while starting cluster: com.tangosol.net.RequestTimeoutException: Timeout
during service start: ServiceInfo(Id=0, Name=Cluster, Type=Cluster
MemberSet=MasterMemberSet(
ThisMember=null
OldestMember=null
ActualMemberSet=MemberSet(Size=0
)
MemberId|ServiceVersion|ServiceJoined|MemberState
RecycleMillis=1200000
RecycleSet=MemberSet(Size=0
)
)
)
at
com.tangosol.coherence.component.util.daemon.queueProcessor.service.Grid.onStartupTimeou
t(Grid.CDB:3)

at
com.tangosol.coherence.component.util.daemon.queueProcessor.Service.start(Service.CDB:28)

at
com.tangosol.coherence.component.util.daemon.queueProcessor.service.Grid.start(Grid.CDB:6
)
```

Solution

This is a known issue. In some of the environments, the Access Policy Manager Server that is not running on the same host as the WebLogic Administration Server is unable to start the coherence cluster service, which results in the oam application deployment to be in failed state. To solve this issue, you must create a server instance for the effected Access Policy Manager Server by completing the following steps:

1. Log in to the OAM console using the following URL:

   `http://`*`iadadmin.example.com`*`/oamconsole`

   Log in as the Access Manager administration user you created when you prepared the ID Store. For example, `oamadmin`.

2. Click **Configuration**.
3. Click **Server Instances** from the configuration launch pad.

4. Click a new server instance for the Access Policy Manager WebLogic Managed Server, that is not running on the same machine as the IAMAccessDomain Admin Server. For example:
   - Name: WLS_AMA2
   - Port: 14150
   - Host: OAMHOST2 (For consolidated topology, the host will be IAMHOST2)

   > **Note:**
   >
   > Provide the OAM Proxy details similar to the server instance for WLS_OAM.

5. Click **Apply**.

Errors in log File when Starting OAM Servers

Problem

When you start the OAM Servers, errors similar to the following are seen in the log files which causes LCM heath check module to fail:

```
[wls_oam1] [TRACE:16] [] [oracle.oam.config] [tid: DistributedCacheWorker:4] [userId:
<anonymous>] [ecid:
0000LGmRJqxB9DE5N7P5ie1N5mOd000004,1:16514] [APP: oam_server#11.1.2.0.0] [SRC_CLASS:
oracle.security.am.admin.config.util.MapUtil] [SRC_METHOD:
getDefaultedStringValue] property not found at path:[Ljava.lang.String;@43537067
Defaulting to value:,
[2016-04-20T06:55:39.982+00:00] [wls_oam1] [TRACE:16] [] [oracle.oam.config] [tid:
DistributedCacheWorker:4] [userId: <anonymous>] [ecid:
0000LGmRJqxB9DE5N7P5ie1N5mOd000004,1:16514] [APP: oam_server#11.1.2.0.0] [SRC_CLASS:
oracle.security.am.admin.config.util.MapUtil] [SRC_METHOD: getStringValue] THROW[[
oracle.security.am.admin.config.ConfigurationException: Cannot get java.lang.String
value from configuration for key ResponseEscapeChar. Object null found.
at
oracle.security.am.admin.config.util.MapUtil.handleFailedAttributeAccess(MapUtil.java:447
)
at oracle.security.am.admin.config.util.MapUtil.getStringValue(MapUtil.java:130)
at oracle.security.am.admin.config.util.MapUtil.getDefaultedStringValue(MapUtil.java:147)
at
oracle.security.am.engines.common.identity.provider.util.IdStoreConfig.initializeConfig(I
dStoreConfig.java:76)
at
oracle.security.am.engines.common.identity.provider.util.IdStoreConfig.<init>(IdStoreConf
ig.java:69)
at
oracle.security.am.engines.common.identity.provider.util.IdStoreConfig.getConfig(IdStoreC
onfig.java:128)
at
oracle.security.am.engines.common.identity.util.OAMUserAttribute.getStringValue(OAMUserAt
tribute.java:76)
at
oracle.security.am.engines.common.identity.util.OAMUserAttribute.toString(OAMUserAttribut
e.java:114)
at java.lang.String.valueOf(String.java:2849)
at java.lang.StringBuilder.append(StringBuilder.java:128)
at java.util.AbstractMap.toString(AbstractMap.java:523)
at java.lang.String.valueOf(String.java:2849)
at java.lang.StringBuilder.append(StringBuilder.java:128)
at
oracle.security.am.engines.common.identity.util.OAMIdentity.toString(OAMIdentity.java:678
)
at java.lang.String.valueOf(String.java:2849)
at java.lang.StringBuilder.append(StringBuilder.java:128)
at oracle.security.am.engines.sso.SSOSubject.toString(SSOSubject.java:238)
at java.lang.String.valueOf(String.java:2849)
at java.lang.StringBuilder.append(StringBuilder.java:128)
at oracle.security.am.engines.sme.impl.SessionImpl.toString(SessionImpl.java:629)
at java.lang.String.valueOf(String.java:2849)
at java.lang.StringBuilder.append(StringBuilder.java:128)
at
oracle.security.am.engines.sme.mapimpl.db.DbOraSmeStore.loadSession(DbOraSmeStore.java:17
05)
at
oracle.security.am.engines.sme.mapimpl.db.DbOraSmeStore.loadSession(DbOraSmeStore.java:16
91)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
at java.lang.reflect.Method.invoke(Method.java:606)
at
oracle.security.am.foundation.mapimpl.coherence.store.DataConnectionUtility.invokeSqlOper
ationWithRetries(DataConnectionUtility.java:275)
at oracle.security.am.engines.sme.mapimpl.db.DbOraSmeStore.load(DbOraSmeStore.java:1284)
at
com.tangosol.net.cache.ReadWriteBackingMap$CacheStoreWrapper.loadInternal(ReadWriteBackin
gMap.java:5676)
at
com.tangosol.net.cache.ReadWriteBackingMap$StoreWrapper.load(ReadWriteBackingMap.java:475
4)
at com.tangosol.net.cache.ReadWriteBackingMap.get(ReadWriteBackingMap.java:717)
at
com.tangosol.coherence.component.util.daemon.queueProcessor.service.grid.partitionedServi
ce.PartitionedCache$Storage.get(PartitionedCache.CDB:10)
at
com.tangosol.coherence.component.util.daemon.queueProcessor.service.grid.partitionedServi
ce.PartitionedCache.onGetRequest(PartitionedCache.CDB:23)
at
com.tangosol.coherence.component.util.daemon.queueProcessor.service.grid.partitionedServi
ce.PartitionedCache$GetRequest.run(PartitionedCache.CDB:1)
at com.tangosol.coherence.component.util.DaemonPool$WrapperTask.run(DaemonPool.CDB:1)
at com.tangosol.coherence.component.util.DaemonPool$WrapperTask.run(DaemonPool.CDB:32)
at com.tangosol.coherence.component.util.DaemonPool$Daemon.onNotify(DaemonPool.CDB:66)
at com.tangosol.coherence.component.util.Daemon.run(Daemon.CDB:42)
at java.lang.Thread.run(Thread.java:745)
]]
```

Solution

This occurs when OAM servers cannot communicate with each other using the coherence port. This is often caused by iptables. The workaround for this issue is as follows:

1. Edit the file `/etc/sysconfig/iptables` on both OAMHOST1 and OAMHOST2 and add the following line:

   ```
   # Generated by iptables-save v1.4.7 on Tue Apr 19 10:02:45 2016
   *filter
   :INPUT ACCEPT [593:243587]
   :FORWARD ACCEPT [0:0]
   :OUTPUT ACCEPT [614:423013]
   -A INPUT -p tcp -m state --state NEW -m tcp --dport 9095 -j ACCEPT
   ```

```
-A INPUT -p tcp -m state --state NEW -m tcp --dport 9097 -j ACCEPT
COMMIT
```

In the above set of lines, `9095` and `9097` are the coherence ports being used.

2. Save the file and restart the servers.

Too Many Redirects Error in Browser

Problem

When navigating from one application to another that uses the same OAM for SSO, you get a redirection error in the web browser. There are two different configurations to validate.

Solution 1:

1. Log in to the OAM Console at `iadadmin.example.com/oamconsole`.
2. From the Launch Pad, click the **Agents** icon.
3. In the resulting window > **Webgates** tab, click **search**. No search parameters need to be input.
4. In the search results, click the IAMSuiteAgent link.
5. Ensure that the Primary Cookie Domain is set to the domain that is used for the `login.example.com` domain. For example: example.com.
6. Restart all WebGate OHS instances.

Solution 2:

Ensure that the date and time on all OHS and OAM servers are within 60 seconds of each other. If they are not:

1. Ensure that the NTP setting are the same and valid on all OHS and OAM hosts.
2. Start or restart the *ntpd* service on all hosts.
3. Restart all WebGate OHS instances, the OAM domain AdminServer, and all Managed Servers.

Troubleshooting Oracle Identity Governance

Learn about some of the common problems that may arise with Oracle Identity Manager and the actions you can take to resolve the problem.

- OIM Bootstrap Process Fails
- java.io.FileNotFoundException When Running Oracle Identity Governance Configuration
- ResourceConnectionValidationxception When Creating User in Oracle Identity Governance
- Oracle Identity Manager Reconciliation Jobs Fail
- OIM Reconciliation Jobs Fail When Running Against Oracle Unified Directory
- Cannot Open Reports from OIM Self Service Console
- Pending Violations Not Displaying the Correct List

OIM Bootstrap Process Fails

Problem

The OIM Bootstrap process fails after deploying composites. The error appears as follows:

```
Deployment of SOA Composites :-/<INSTALL_LOCATION>/Oracle_Home/idm/server/workflows/
composites/scajars/sca_DefaultRequestApproval_rev6.0.jar is successful>
<Jun 12, 2018 4:20:26,136 PM CEST> <Info> <oracle.iam.OIMPostConfigManager> <BEA-000000>
<updating feature:DEPLOYSOACOMPOSITESwith state :COMPLETEwith executionTime190108>
java.sql.SQLException: Connection closed
```

This is caused by a performance issue.

Solution

To resolve the issue temporarily, increase the inactivity timeouts on the following data sources:

- `oimJMSStoreDS`
- `oimOperationsDB`

The settings can be restored to their original values after the upgrade is complete.

1. Log in to the WebLogic Server Administration Console.
2. Click **Lock and Edit**.
3. Click **Services**, **Data Sources**, and then select the <*Data source name*>.
4. Click the **Connection Pool** tab.
5. Under the Advanced section, increase the value of **Inactive Connection Timeout**.
6. Save and activate the changes.
7. Restart the OIM Managed Server.

java.io.FileNotFoundException When Running Oracle Identity Governance Configuration

Problem

The following content was added to address bug 12390838

When you run Oracle Identity Manager configuration, the error `java.io.FileNotFoundException: soaconfigplan.xml (Permission denied`) may appear and Oracle Identity Manager configuration might fail.

Solution

To workaround this issue:

1. Delete the file `/tmp/soaconfigplan.xml`.
2. Start the configuration again (*`IGD_ORACLE_HOME`*`/bin/config.sh`).

ResourceConnectionValidationxception When Creating User in Oracle Identity Governance

Problem

The following content was added to address bug 9816870

If you are creating a user in Oracle Identity Manager (by logging into Oracle Identity Manager System Administration Console, clicking the Administration tab, clicking the **Create User** link, entering the required information in the fields, and clicking **Save**) in an active-active Oracle Identity Manager configuration, and the Oracle Identity Manager server that is handling the request fails, you may see a "ResourceConnectionValidationxception" in the Oracle Identity Manager log file, similar to:

```
[2010-06-14T15:14:48.738-07:00] [oim_server2] [ERROR] [] [XELLERATE.SERVER]
[tid: [ACTIVE].ExecuteThread: '0' for queue: 'weblogic.kernel.Default
(self-tuning)'] [userId: xelsysadm] [ecid:
004YGJGmYrtEkJV6u3M6UH00073A0005EI,0:1] [APP: oim#11.1.1.3.0] [dcid:
12eb0f9c6e8796f4:-785b18b3:12938857792:-7ffd-0000000000000037] [URI:
/admin/faces/pages/Admin.jspx] Class/Method:
PooledResourceConnection/heartbeat encounter some problems: Operation timed
out[[
com.oracle.oim.gcp.exceptions.ResourceConnectionValidationxception: Operation
timed out
        at
oracle.iam.ldapsync.impl.repository.LDAPConnection.heartbeat(LDAPConnection.ja
va:162)
        at
com.oracle.oim.gcp.ucp.PooledResourceConnection.heartbeat(PooledResourceConnec
tion.java:52)
         .
         .
         .
```

Solution

Despite this exception, the user is created correctly.

Oracle Identity Manager Reconciliation Jobs Fail

Problem

Oracle Identity Manager reconciliation jobs fail, or one of the following messages is seen in the log files:

- Error-1

  ```
  LDAP Error 53 : [LDAP: error code 53 - Full resync required. Reason: The provided
  cookie is older than the start of historical in the server for the replicated
  domain : dc=example,dc=com]
  ```

- Error-2

  ```
  LDAP: error code 53 - Invalid syntax of the provided cookie
  ```

This error is caused by the data in the Oracle Unified Directory change log cookie expiring because Oracle Unified Directory has not been written to for a certain amount of time.

Solution

1. Open a browser and go to the following location:

   ```
   http://igdadmin.example.com/sysadmin
   ```

2. Log in a as `xelsysadm` using the *`COMMON_IDM_PASSWORD`*.
3. Under **System Management**, click **Scheduler**.
4. Under **Search Scheduled Jobs**, enter `LDAP *` (there is a space before *) and hit **Enter**.
5. For each job in the search results, click on the job name on the left, then click **Disable** on the right.

   Do this for all jobs. If the job is already disabled do nothing.

6. Run the following commands on LDAPHOST1:

   ```
   cd LDAP_ORACLE_INSTANCE/OUD/bin
   ./ldapsearch -h LDAPHOST1 -p 1389 -D "cn=oudadmin" -b "" -s base "objectclass=*"
   lastExternalChangelogCookie

   Password for user 'cn=oudadmin': <OudAdminPwd>
   dn: lastExternalChangelogCookie: dc=example,dc=com:00000140c682473c263600000862;
   ```

   Copy the output string that follows `lastExternalChangelogCookie:`. This value is required in the next step. For example,

   ```
   dc=example,dc=com:00000140c682473c263600000862;
   ```

   The Hex portion must be 28 characters long. If this value has more than one Hex portion then separate the 28char portions with spaces. For example:

   ```
   dc=example,dc=com:00000140c4ceb0c07a8d00000043 00000140c52bd0b9104200000042
   00000140c52bd0ba17b9000002ac 00000140c3b290b076040000012c;
   ```

7. Run each of the following LDAP reconciliation jobs once to reset the last change number.:

   - LDAP Role Delete Reconciliation
   - LDAP User Delete Reconciliation
   - LDAP Role Create and Update Reconciliation
   - LDAP User Create and Update Reconciliation
   - LDAP Role Hierarchy Reconciliation
   - LDAP Role Membership Reconciliation

   To run the jobs:

   a. Login to the OIM System Administration Console as the user `xelsysadm`.

   b. Under **System Configuration**, click **Scheduler**.

   c. Under **Search Scheduled Jobs**, enter `LDAP *` (there is a space before *) and hit **Enter**.

   d. Click on the job to be run.

   e. Set the parameter **Last Change Number** to the value obtained in step 6.

      For example:

      ```
      dc=example,dc=com:00000140c4ceb0c07a8d00000043 00000140c52bd0b9104200000042
      00000140c52bd0ba17b9000002ac 00000140c3b290b076040000012c;
      ```

   f. Click **Run Now**.

   g. Repeat for each of the jobs in the list at the beginning of this step.

8. For each incremental recon job whose last changelog number has been reset, execute the job and check that the job now completes successfully.

9. After the job runs successfully, re-enable periodic running of the jobs according to your requirements.

If the error appears again after the incremental jobs have been re-enabled and run successfully ("Full resync required. Reason: The provided cookie is older..."), then increase the OUD cookie retention time. Although there is no hard and fast rule as to what this value should be, it should be long enough to avoid the issue, but small enough to avoid unnecessary resource consumption on OUD. One or two weeks should suffice. Run the following command on each OUD instance to increase the retention time to two weeks:

```
cd OUD_ORACLE_INSTANCE/bin

./dsconfig set-replication-server-prop --provider-name "Multimaster Synchronization" --
set replication-purge-delay:2w -D cn=oudadmin --trustAll -p 4444 -h LDAPHOSTn

Password for user 'cn=oudadmin':  <OudAdminPswd>
Enter choice [f]: f
```

OIM Reconciliation Jobs Fail When Running Against Oracle Unified Directory

Problem

Reconciliation jobs fail when running against Oracle Unified Directory (OUD). The following error is seen in the OIM WebLogic Server logs:

```
LDAP: error code 53 - Invalid syntax of the provided cookie
```

Solution

Perform the workaround described in Oracle Identity Manager Reconciliation Jobs Fail. If this workaround does not resolve the issue, try the following solution:

On each OIMHOST, update the `IGD_MSERVER_HOME/config/fmwconfig/ovd/oim/adapters.os_xml` file with the following parameter:

```
<param name="eclCookie" value="false"/>
```

Restart the OIM and SOA Managed Servers.

Cannot Open Reports from OIM Self Service Console

Problem

The reports cannot be opened from OIM Self Service Console.

Solution

When you enable the Identity Auditor feature in OIM, do the following configuration changes for the OIM-BI Publisher integration to work fine:

1. Log in to the IAMGovernanceDomain Enterprise Management Console.
2. Open the system MBean browser and update the MBean `"oracle.iam:Location=wls_oim1,name=Discovery,type=XMLConfig.DiscoveryConfig,XMLConfig=Config,Application=oim,ApplicationVersion=11.1.2.0.0"` with Value as `http://igdadmin.example.com/`.

   Here, *`igdadmin.example.com`* is the Governance Domain admin Load balancer URL.

Pending Violations Not Displaying the Correct List

Problem

When viewing the pending violations list, you may see entries that are missing or entries that do not belong to the list.

Solution

If you encounter this issue, a restart of the OIG domain usually resolves it. If the issue is not resolved, raise a Service Request (SR) with Oracle Support.

Troubleshooting Oracle SOA Suite

Learn about the transaction timeout error that may arise with Oracle SOA Suite and the action you can take to resolve the problem.

- Transaction Timeout Error

Transaction Timeout Error

Problem

The following transaction timeout error appears in the log:

```
Internal Exception: java.sql.SQLException: Unexpected exception while enlisting
 XAConnection java.sql.SQLException: XA error: XAResource.XAER_NOTA start()
failed on resource 'SOADataSource_soaedg_domain': XAER_NOTA : The XID
is not valid
```

Solution

Check your transaction timeout settings, and be sure that the JTA transaction time out is less than the DataSource XA Transaction Timeout, which is less than the `distributed_lock_timeout` (at the database).

With the out of the box configuration, the SOA data sources do not set XA timeout to any value. The `Set XA Transaction Timeout` configuration parameter is unchecked in the WebLogic Server Administration Console. In this case, the data sources use the domain level JTA timeout which is set to `30`. Also, the default `distributed_lock_timeout` value for the database is `60`. As a result, the SOA configuration works correctly for any system where transactions are expected to have lower life expectancy than such values. Adjust these values according to the transaction times your specific operations are expected to take.

Troubleshooting Integration OIGOAMIntegration.sh-configureLDAPConnector

Learn about the error you may encounter during the inegration process and the solution to fix this error.

Problem

The following content was added to address bug 27567130

Whilst running configureLDAPConnector, you see the following error message:

```
2018-02-19 06:54:05] LDAPConnectorConfigTool.configureLDAPConnector:
exception: java.lang.reflect.UndeclaredThrowableException  [2018-02-19
06:54:05] javax.management.InstanceNotFoundException: Unable to  contact
MBeanServer for
oracle.iam:Location=oim_server1,name=SSOIntegrationMXBean,type=IAMAppRuntimeMB
  ean,Application=oim  at weblogic.utils.StackTraceDisabled.unknownMethod()
```

Solution

This is caused by the OIM Managed Server being called something other than `oim_server1`. This can be recovered by executing the following workaround.

Ensure that your OIM Managed Server is running.

1. Log in to Oracle Fusion Middleware control using the following URL: `http://igdadmin.example.com/em`.
2. Start the System Mbean Browser by selecting Weblogic Domain and then clicking on System MBean browser.
3. Click on find and enter the Mbean name **SSOIntegrationMXBean** .
4. Click **Search**.
5. When the MBean is found, click **Operations > addContainerRules** .
6. Enter the following information:

   ```
   Oracle_Home set to the value of IGD_ORACLE_HOME dirType. set to OUD
   userContainer set to
   cn=users,
   dc=example,
   dc=com
   roleContatiner set to cn=groups,
   dc=example,dc=com
   ```

7. Click **Invoke** button.

General Troubleshooting

Learn about the error you may encounter when starting the Managed Server from the WebLogic Console and the resolution to fix the error.

- Cannot Start Managed Server from WebLogic Console

Cannot Start Managed Server from WebLogic Console

Problem

When you start a Managed Server from the WebLogic Console, the following error is shown:

```
. For server WLS_BI1, the Node Manager associated with machine OIMHOST1 is not reachable.
. All of the servers selected are currently in a state which is incompatible with this
operation or are not associated with a running Node Manager or you are not authorized to
perform the action requested. No action will be performed.
```

Solution 1

Check if the Node Manager is started on the target host. If not, start it.

Solution 2

Verify that the domain is listed in the file `nodemanager.domains`, which is located in the directory *`SHARED_CONFIG_DIR`*`/nodemanger/hostname`. If not, do the following:

1. Start the WebLogic Scripting Tool (WLST) by running the following command from the location *`ORACLE_HOME`*`/oracle_common/common/bin/`:

   ```
   ./wlst.sh
   ```

2. Connect to the domain you wish to add by running the following command:

   ```
   connect('weblogic_user','password','t3://ADMINVHN:AdminPort')
   ```

   In this command:

   weblogic_user is the WebLogic Administration user. For example, `weblogic` or `weblogic_idmw`.

   password is the password of the WebLogic Administration user.

   ADMINVHN is the Virtual host name of the Administration Server. For example, `IGDADMINVHN` or `IADADMINVHN`.

   `adminPort` is the port on which the Administration Server is running. For example, `7101`.

   Sample Command:

   ```
   connect('weblogic_idm','<password>','t3://IGDADMINVHN.example.com:7001')
   ```

3. Enrol the domain using the following command:

   ```
   nmEnroll(domainDir=absolute_path_to_the_domain,nm_Home=absolute_path_to_the_no
   demanager_home)
   ```

   For example:

   ```
   nmEnroll(domainDir='/u02/private/oracle/config/domains/
   IAMGovernanceDomain/',nmHome='/u01/oracle/config/nodemanger/hostname)')
   ```

   > **Note:**
   >
   > For Managed Servers, the domain home should always be specified as the local Managed Server directory.