# Oracle® Fusion Middleware

Enterprise Deployment Guide for Oracle Business Intelligence

Release 12.2.1

**E57383-04**

February 2016

Describes how to install and configure Oracle Fusion Middleware components in an enterprise deployment.

ORACLE®

Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence, Release 12.2.1

E57383-04



Primary Authors: Susan Kornberg (MAA Engineer), Phil Stubbs (Writer), Srinivasan Varadarajan (Quality Assurance)

Contributing Authors: Fermin Castro, Peter LaQuerre

Contributors: Joe Paul, Michael Rhys, Allwarappan Sundararaj, Chakravarthy Tenneti, Bonnie Vaughan

# Contents

## Part II Preparing for an Enterprise Deployment

## 11 Configuring the Web Tier for an Enterprise Deployment

## 12 Scaling Out Oracle Business Intelligence



# Part IV Common Configuration and Management Procedures for an Enterprise Deployment

## 13 Common Configuration and Management Tasks for an Enterprise Deployment

## A Using Multi Data Sources with Oracle RAC

# Preface

This guide explains how to install, configure, and manage a highly available Oracle Fusion Middleware enterprise deployment. For more information, see About the Enterprise Deployment Guide.

**See Also:**

Audience

Conventions

## 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 more specialized administrators, such as database administrators (DBAs) and network administrators, where applicable.

## 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.

---

# Part I

## Understanding an Enterprise Deployment

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

**See Also:**

Enterprise Deployment Overview

Understanding a Typical Enterprise Deployment

Understanding the Business Intelligence Enterprise Deployment Topology

# 1

# Enterprise Deployment Overview

This chapter introduces the concept of an Oracle Fusion Middleware enterprise deployment.

It also provides information on when to use the Enterprise Deployment guide.

**See Also:**

Understanding an Enterprise Deployment

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. This resulting environment is known as an **enterprise deployment topology**.

When to Use the Enterprise Deployment Guide
This guide describes one of 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.

## 1.1 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. This resulting environment is known as an **enterprise deployment topology**.

By 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 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 related Oracle documentation or by working with Oracle Support.

## 1.2 When to Use the Enterprise Deployment Guide

This guide describes one of 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 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 Business Intelligence, see:

  – *Installing and Configuring Oracle Business Intelligence*

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

# 2

# Understanding a Typical Enterprise Deployment

This chapter describes the general characteristics of a typical Oracle Fusion Middleware enterprise deployment. You can apply the concepts here to any product-specific enterprise deployment.

This chapter provides information on Enterprise Deployment Topology diagram.

**See Also:**

Understanding an Enterprise Deployment

Diagram of a Typical Enterprise Deployment
This section illustrates a typical enterprise deployment, including the Web tier, application tier, and data tier.

Understanding the Typical Enterprise Deployment Topology Diagram
This section provides a detailed description of the typical enterprise topology diagram.

## 2.1 Diagram of a Typical Enterprise Deployment

This section illustrates a typical enterprise deployment, including the Web tier, application tier, and data tier.

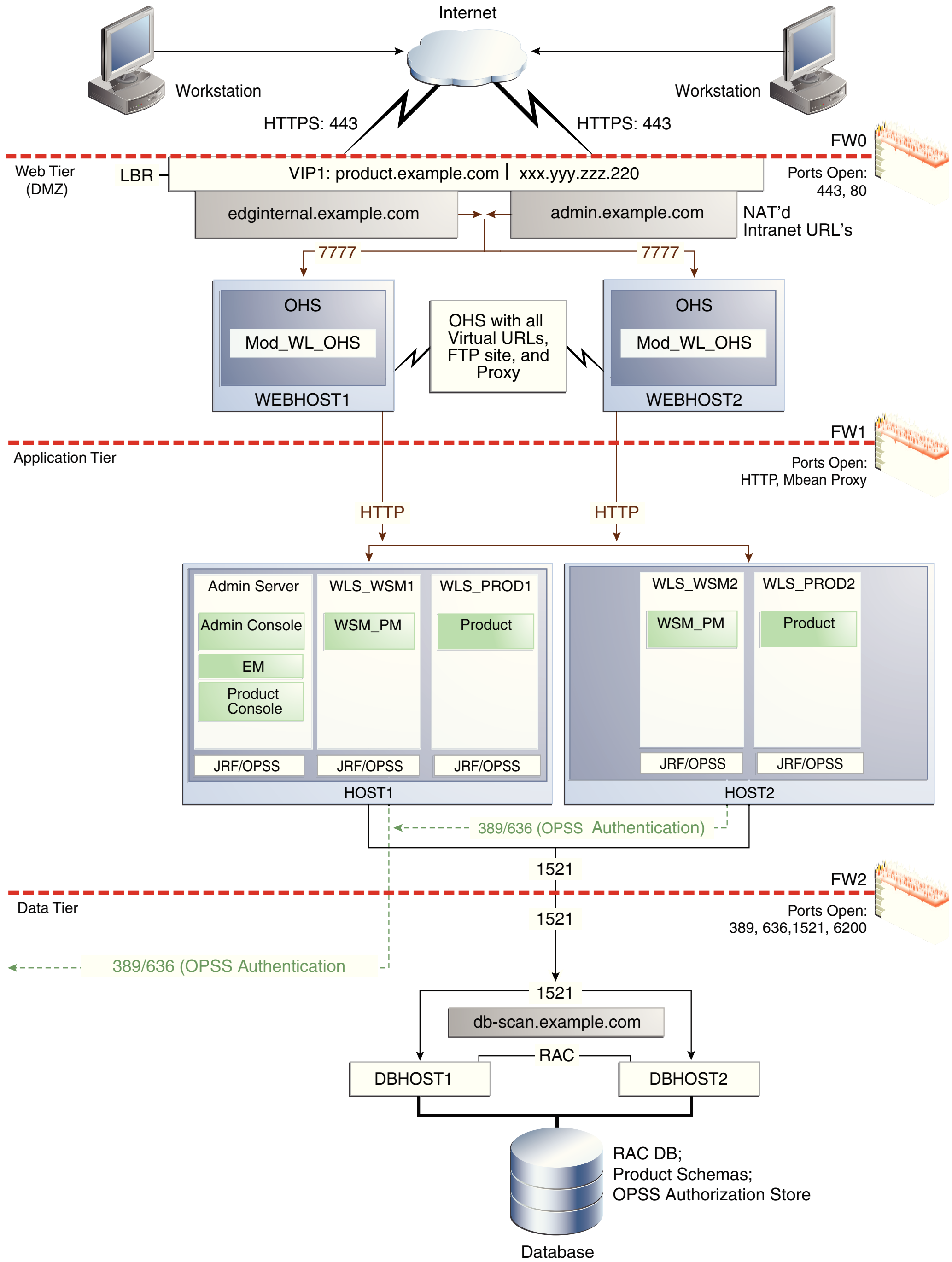
All Oracle Fusion Middleware enterprise deployments are designed to demonstrate the best practices for installing and configuring an Oracle Fusion Middleware production environment.

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 Understanding the Typical Enterprise Deployment Topology Diagram.

***Figure 2-1 Typical Enterprise Deployment Topology Diagram***




## 2.2 Understanding the Typical Enterprise Deployment Topology Diagram

This section provides a detailed description of the typical enterprise topology diagram.

**See Also:**

Understanding the Firewalls and Zones of a Typical Enterprise Deployment

Understanding the Elements of a Typical Enterprise Deployment Topology

Receiving Requests Through Hardware Load Balancer

Understanding the Web Tier

Understanding the Application Tier

About the Data Tier

## 2.2.1 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 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 will 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 the 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 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 you must open for a specific Oracle Fusion Middleware enterprise deployment topology, see the chapter that describes the topology you want to implement, or refer to the *Enterprise Deployment Workbook* for the topology you are implement. For more information, see Using the Enterprise Deployment Workbook .

## 2.2.2 Understanding 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 are performing 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 (via an external identity store and a single sign-on server) and then route the HTTP requests to the Oracle Fusion Middleware products and components running in the Application tier.

  The Web server instances also host static Web content that does not require 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 Server 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.

## 2.2.3 Receiving Requests Through Hardware Load Balancer

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

**See Also:**

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

### 2.2.3.1 Purpose of the Hardware Load Balancer (LBR)

The following topics describe the types of requests handled by the hardware load balancer in an enterprise deployment.

**See Also:**

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

#### 2.2.3.1.1 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 will be 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.

#### 2.2.3.1.2 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, using a unique virtual host name.

### 2.2.3.2 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. As shown in the diagram, 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 you have deployed and the custom applications available on this server. The load balancer then routes these requests (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.

- *`product`*`internal.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, we prefix it 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 the complete set of virtual server names you must define for your topology, see the the chapter that describes the product-specific topology.

#### 2.2.3.3 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. For more information, see Configuring Virtual Hosts on the Hardware Load Balancer.

### 2.2.4 Understanding the Web Tier

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

**See Also:**

Benefits of Using Oracle HTTP Server Instances to Route Requests

Alternatives to Using Oracle HTTP Server in the Web Tier

Configuration of Oracle HTTP Server in the Web Tier

About Mod_WL_OHS

#### 2.2.4.1 Benefits of Using Oracle HTTP Server Instances 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 does provide several advantages, which is why it is recommended as part of the reference topology:

- 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 FTP services, which are required for some enterprise deployments, as well as the ability to create virtual hosts and proxies via the Oracle HTTP Server configuration files.

- Oracle HTTP Server provides HTTP redirection over and above what 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 via 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, using Oracle Access Manager, which is part of the Oracle Identity and Access Management family of products.

- Oracle HTTP Server provides support for WebSocket connections deployed within WebLogic Server.

For more information about Oracle HTTP Server, see Introduction to Oracle HTTP Server in *Administrator's Guide for Oracle HTTP Server*.

#### 2.2.4.2 Alternatives to Using Oracle HTTP Server in the Web Tier

Although Oracle HTTP Server 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 OHS).

  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.

#### 2.2.4.3 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 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.

For more information, see "Understanding Oracle HTTP Server Installation Options" in *Installing and Configuring Oracle HTTP Server*.

#### 2.2.4.4 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.

For more information, see "Overview of Web Server Proxy Plug-Ins 12.1.3" in *Using Oracle WebLogic Server Proxy Plug-Ins 12.1.3*.

## 2.2.5 Understanding 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.

**See Also:**

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

Understanding OPSS and Requests to the Authentication and Authorization Stores

### 2.2.5.1 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 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.

#### 2.2.5.2 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.

#### 2.2.5.3 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.

#### 2.2.5.4 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 controlling the Administration Server uses the listen address of the virtual host name created for the Administration Server. The Node Manager controlling 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 Domain 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 machine.

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.

### 2.2.5.5 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 benefits of each protocol.

#### Benefits 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.

#### Benefits 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 router(s) 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 behave 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 will behave better in those cases.

For more information see the following resources:

- "Configuring Multicast Messaging for WebLogic Server Clusters" in the *High Availability Guide*
- "One-to-Many Communication Using Unicast" in *Administering Clusters for Oracle WebLogic Server*.

#### 2.2.5.6 Understanding 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 Internet Directory (OID) or 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 OPSS Policy Model"

### 2.2.6 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 Business Intelligence 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; for more information, see the *Oracle Data Guard Concepts and Administration*
- Oracle RAC One Node; for more information, see "Overview of Oracle RAC One Node" in the *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 the *High Availability Guide*.

# 3

# Understanding the Business Intelligence Enterprise Deployment Topology

The following topics describe the Oracle Business Intelligence enterprise deployment topologies.

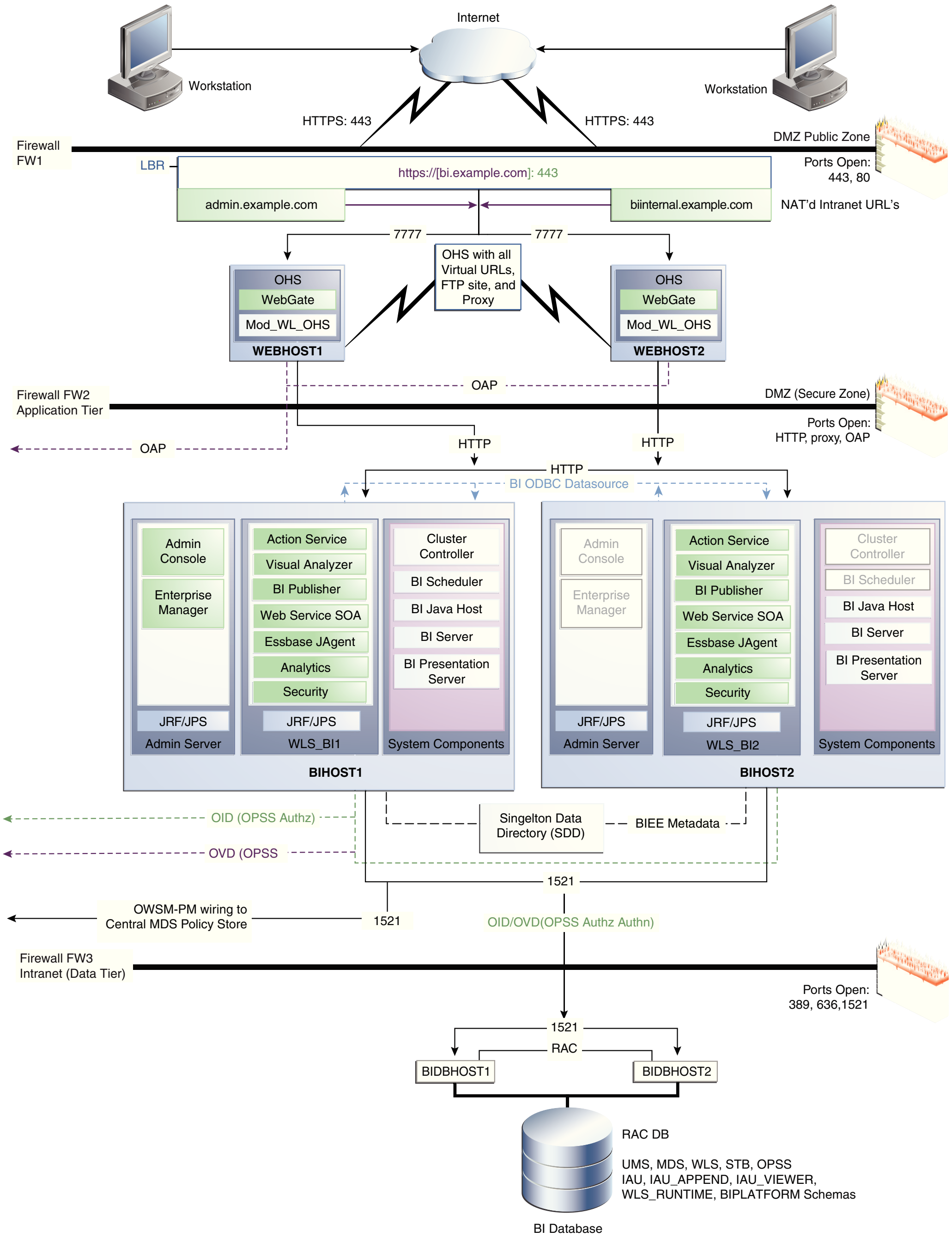
These topologies represent specific reference implementations of the concepts described in Understanding a Typical Enterprise Deployment.

**See Also:**

Understanding an Enterprise Deployment

## 3.1 Diagram of the Primary Business Intelligence Enterprise Topology

This diagram shows the primary Oracle Business Intelligence enterprise deployment topology.




## 3.2 Understanding the Primary Business Intelligence Topology Diagrams

This section provides information about the elements that are unique to the primary topology.

Most of the elements of Oracle Business Intelligence topologies represent standard features of any enterprise topology that follows the Oracle-recommended best practices. These elements are described in detail in Understanding a Typical Enterprise Deployment.

Before you review the information here, it is assumed you have reviewed the information in Understanding a Typical Enterprise Deployment and that you are familiar with the general concepts of an enterprise deployment topology.

See the following sections for information about the elements that are unique to the topology described in this chapter.

**See Also:**

Summary of Business Intelligence Load Balancer Virtual Server Names

Summary of the Managed Servers and Cluster on the Business Intelligence Application Tier

### 3.2.1 Summary of Business Intelligence 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.

For information about the purpose of each of these server names, see Summary of the Typical Load Balancer Virtual Server Names.

The following virtual server names are recognized by the hardware load balancer in Oracle Business Intelligence topologies:

- `bi.example.com` - This virtual server name is used for all incoming traffic. It acts as the access point for all HTTP traffic to the runtime Business Intelligence 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:

  ```
  bi.example.com:443
  ```

- `biinternal.example.com` - This virtual server name is for internal communications between the application tier components only and is not exposed to the Internet.

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

  ```
  biinternal.example.com:80
  ```

- `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.

Use instructions later in this guide to perform the following tasks:

- Configure the hardware load balancer to recognize and route requests to the virtual host names
- Configure the Oracle HTTP Server instances on the Web Tier to recognize and properly route requests to these virtual host names to the correct host computers.

### 3.2.2 Summary of the Managed Servers and Cluster on the Business Intelligence Application Tier

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

Depending upon the topology you select, the Oracle WebLogic Server domain for the domain consists of the cluster shown in Summary of the Managed Servers and Clusters on the Business Intelligence Application Tier. This cluster functions as active-active high availability configurations.

*Table 3-1 Summary of the Cluster in the Oracle Business Intelligence Enterprise Deployment Topology*

| Cluster | Managed Servers |
|---|---|
| Oracle Business Intelligence | WLS_BI1, WLS_BI2 |

## 3.3 Flow Charts and Roadmaps for Implementing the Primary Business Intelligence Enterprise Topologies

This section summarizes the high-level steps you must perform to install and configure the enterprise topology described in this chapter.

**See Also:**

Flow Chart of the Steps to Install and Configure the Primary Business Intelligence Enterprise Topologies

Roadmap Table for Planning and Preparing for an Enterprise Deployment

Roadmap Table for Configuring the Business Intelligence Enterprise Topology

### 3.3.1 Flow Chart of the Steps to Install and Configure the Primary Business Intelligence Enterprise Topologies

Flow Chart of the Steps to Install and Configure the Primary Business Intelligence Enterprise Topologies shows a flow chart of the steps required to install and configure the primary enterprise deployment topologies described in this chapter. The sections following the flow chart explain each step in the flow chart.

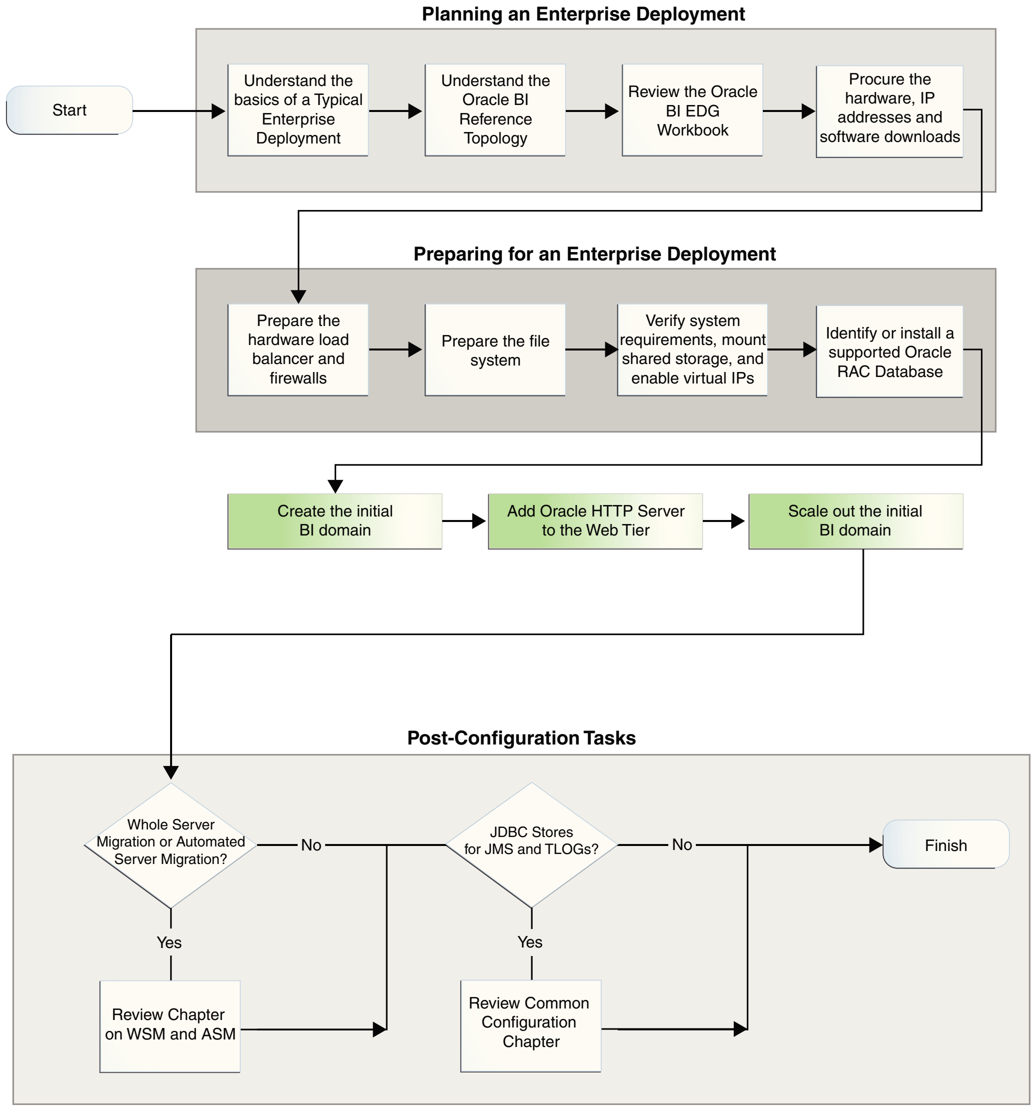
This guide is designed so you can start with a working Business Intelligence domain and then later scale out the domain to add additional capabilities.

This modular approach to building the topology allows you to make strategic decisions, based on your hardware and software resources, as well as the Oracle Business Intelligence features that are most important to your organization.

It also allows you to validate and troubleshoot each individual product or component as they are configured.

This does not imply that configuring multiple products in one Configuration Wizard session is not supported; it is possible to group various extensions like the ones presented in this guide in one Configuration Wizard execution. However, the instructions in this guide focus primarily on the modular approach to building an enterprise deployment.

*Figure 3-1 Flow Chart of the Enterprise Topology Configuration Steps*




## 3.3.2 Roadmap Table for Planning and Preparing for an Enterprise Deployment

The following table describes each of the planning and preparing steps shown in the enterprise topology flow chart.

| Flow Chart Step | More Information |
|---|---|
| Understand the basics of a Typical Enterprise Deployment | Understanding a Typical Enterprise Deployment |
| Understand the specific reference topology for the products you plan to deploy. | Review the product-specific topologies and the description of the topologies, including the virtual servers required and the summary of clusters and Managed Servers recommended for the product-specific deployment. |

| Flow Chart Step | More Information |
| --- | --- |
| Review the Oracle Business Intelligence EDG Workbook | Using the Enterprise Deployment Workbook |
| Procure the hardware, IP addresses and software downloads | Procuring Resources for an Enterprise Deployment |
| Prepare the hardware load balancer and firewalls | Preparing the Load Balancer and Firewalls for an Enterprise Deployment |
| Prepare the file system | Preparing the File System for an Enterprise Deployment |
| Verify system requirements, mount shared storage, and enable virtual IPs | Preparing the Host Computers for an Enterprise Deployment |
| Identify or install a supported Oracle RAC Database | Preparing the Database for an Enterprise Deployment |

## 3.3.3 Roadmap Table for Configuring the Business Intelligence Enterprise Topology

Table 3-2 describes each of the configuration steps required when configuring the topology shown in Diagram of the Primary Business Intelligence Enterprise Topology.

These steps correspond to the steps shown in the flow chart in Flow Chart of the Steps to Install and Configure the Primary Business Intelligence Enterprise Topologies.

***Table 3-2 Roadmap Table for Configuring the Business Intelligence Enterprise Topology***

| Flow Chart Step | More Information |
| --- | --- |
| Create the initial Business Intelligence domain | Creating the Initial BI Domain for an Enterprise Deployment |
| Extend the domain to include the Web Tier | Configuring the Web Tier for an Enterprise Deployment |
| Scale out the initial Business Intelligence domain | Scaling Out Oracle Business Intelligence |

# Part II

# Preparing for an Enterprise Deployment

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

**See Also:**


Using the Enterprise Deployment Workbook

Procuring Resources for an Enterprise Deployment

Preparing the Load Balancer and Firewalls for an Enterprise Deployment

Preparing the File System for an Enterprise Deployment

Preparing the Host Computers for an Enterprise Deployment

Preparing the Database for an Enterprise Deployment

# 4

# Using the Enterprise Deployment Workbook

This chapter introduces the Enterprise Deployment Workbook; it describes how you can use the workbook 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.

**See Also:**



## 4.1 Introduction to the Enterprise Deployment Workbook

This section provides a brief introduction of the enterprise deployment workbook.

The Oracle Fusion Middleware Enterprise Deployment Workbook is a companion document to this guide. It is a spreadsheet that can be 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 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 actual implementation
- Consistency during implementation
- A record of the environment for future use

## 4.2 Typical Use Case for Using the Workbook

This section lists 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 system engineering departments;
- The Administrators and System Integrators who are installing and configuring the software refer to the workbook and the subsequent chapters of this guide to perform the installation and configuration tasks.

## 4.3 Using the Oracle Business Intelligence Enterprise Deployment Workbook

This section provides details for using the enterprise deployment workbook.

The following sections provide an introduction to the location and contents of the Oracle Business Intelligence Enterprise Deployment Workbook:

**See Also:**

Locating the Oracle Business Intelligence Enterprise Deployment Workbook

Understanding the Contents of the Oracle Business Intelligence Enterprise Deployment Workbook

### 4.3.1 Locating the Oracle Business Intelligence Enterprise Deployment Workbook

The Oracle Business Intelligence 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.

### 4.3.2 Understanding the Contents of the Oracle Business Intelligence Enterprise Deployment Workbook

The following sections describe the contents of the Oracle Business Intelligence 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 Enterprise Deployment topologies.

**See Also:**

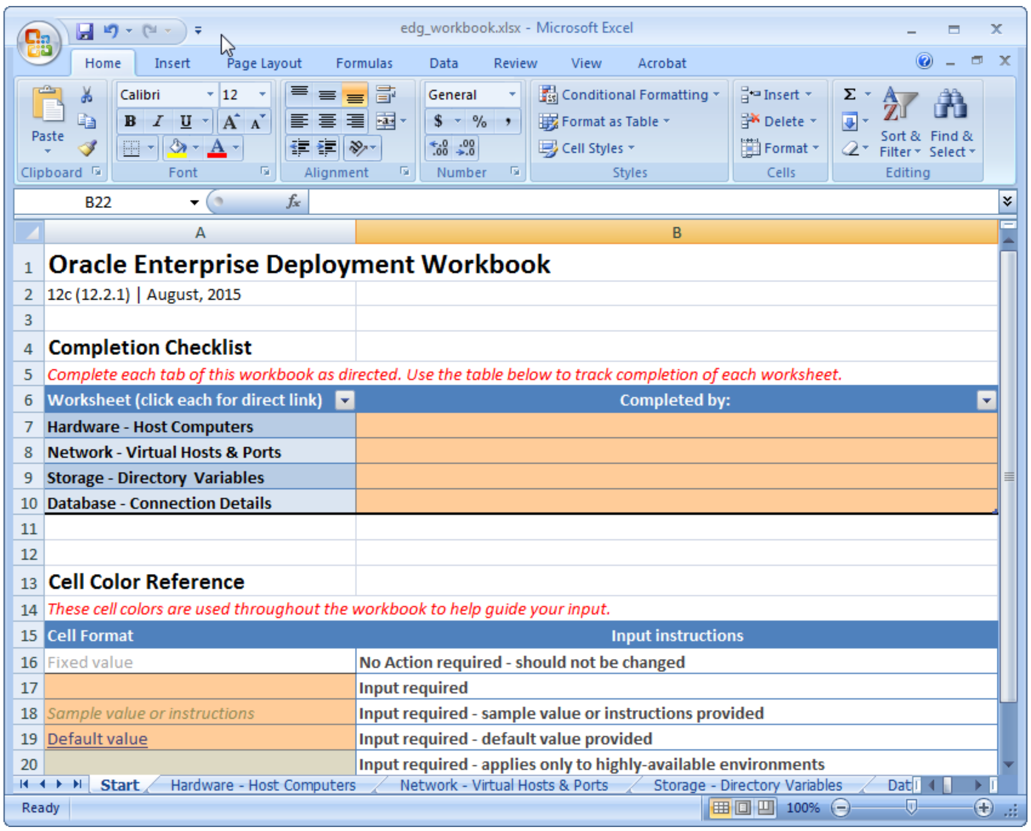
Using the Start Tab

#### 4.3.2.1 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.

The following image shows the Start tab of the spreadsheet.




#### 4.3.2.2 Using the Hardware - Host Computers Tab

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

The reference topologies typically 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 tier. 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**. You can then use the actual host name when any of the abstract names is referenced in this guide.

For example, if a procedure in this guide references BIHOST1, you can then replace the BIHOST1variable with the actual name provided on the Hardware - Host Computers tab of the workbook.

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.

For more information, see Preparing the Host Computers for an Enterprise Deployment.

#### 4.3.2.3 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 you can access the management consoles; the firewalls must also be configured to allow network traffic via specific ports.

Each virtual host, virtual IP address, and each network port serves a distinct purpose in the deployment. For more information, 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 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 Network - Port Numbers table to map the default port values to the actual values used in your specific installation.

#### 4.3.2.4 Using the Storage - Directory Variables Tab

As part of preparing for an enterprise deployment, it is assumed you will be 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 used when you mounted the shared location.

For more information, see Preparing the File System for an Enterprise Deployment.

#### 4.3.2.5 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.

For more information, see Preparing the Database for an Enterprise Deployment.

## 4.4 Who Should Use the Enterprise Deployment Workbook?

The information in the Enterprise Deployment Workbook is divided into categories. Depending on the structure of your organization and roles 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 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

# 5

# Procuring Resources for an Enterprise Deployment

Use the following topics to procure the required hardware, software, and network settings before you begin configuring the Oracle Business Intelligence reference topology.

This chapter provides information on reserving the required IP addresses and identifying and obtaining software downloads for an enterprise deployment.

**See Also:**

Preparing for an Enterprise Deployment

Hardware and Software Requirements for the Enterprise Deployment Topology
This section specifies 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
This section lists the set of IP addresses that you must obtain and reserve before installation and configuration.

Identifying and Obtaining Software Downloads for an Enterprise Deployment
Before you begin installing and configuring the enterprise topology, you should locate and download the software distributions that you will need to implement the topology.

## 5.1 Hardware and Software Requirements for the Enterprise Deployment Topology

This section specifies the hardware load balancer requirements, host computer hardware requirements, and operating system requirements for the enterprise deployment topology.

This section includes the following sections.

**See Also:**

Hardware Load Balancer Requirements
This section lists the desired 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 the Enterprise Deployment Topology
This section provides details about the operating system requirements.

### 5.1.1 Hardware Load Balancer Requirements

This section lists the desired features of the external load balancer.

This enterprise topology uses an external load balancer. This external load balancer should have the following features:

- Ability to load-balance traffic to a pool of real servers through a virtual host name: Clients access services using the virtual host name (instead of 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.
- Virtual servers and port configuration: Ability to configure virtual server names and ports on your external load balancer, and the virtual server names and ports must meet the following requirements:
  – The load balancer should 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.
- Fault-tolerant mode: 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.
- Sticky routing capability: 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 using the equivalent non-SSL protocol (for example, HTTPS to HTTP).

- SSL acceleration (this feature is recommended, but not required for the enterprise topology).
- The ability to route TCP/IP requests; this is a requirement for Oracle SOA Suite for healthcare integration, which uses the Minimum Lower Layer Protocol (MLLP) over TCP.

## 5.1.2 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.

**See Also:**



### 5.1.2.1 General Considerations for Enterprise Deployment Host Computers

This section specifies the general considerations 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 will vary for each application or custom Oracle Business Intelligence system being used.

The information in this chapter provides general guidelines and information that will help 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.

For more information, see Using the Enterprise Deployment Workbook

#### 5.1.2.2 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 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 required operating system libraries and packages.

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

#### 5.1.2.3 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 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 Business Intelligence 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 BIHOST1, as depicted in the reference topologies.

When you are procuring machines, use the information in the **Approximate Top Memory** column as a guide when determining how much physical memory 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 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.

For more information, 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 |
|---|---|---|---|
| Administration Server | 3.5 GB | 3500 | 165 |

| Managed Server, Utility, or Service | Approximate Top Memory | Number of File Descriptors | Operating System Processes and Tasks |
|---|---|---|---|
| WLS_BI | 4.0 GB | 31000 | 130 |
| WLST (connection to the Node Manager) | 1.5 GB | 910 | 20 |
| Configuration Wizard | 1.5 GB | 700 | 20 |
| Node Manager | 1.0 GB | 720 | 15 |
| System components | TBD | TBD | TBD |
| TOTAL | 11.0 GB* | 17000 | 1200 |

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

#### 5.1.2.4 Typical Disk Space Requirements for an Enterprise Deployment

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

For the latest disk space requirements for the Oracle Fusion Middleware 12*c* (12.2.1) products, including the Oracle Business Intelligence products, review the *Oracle Fusion Middleware System Requirements and Specifications*.

In addition, the following table summarizes the disk space typically required for an Oracle Business Intelligence 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<br>n = number of disks, at least 4 (striped as one disk)<br>m = size of the disk (minimum of 30 GB) |
| WEBHOST*n* | 10 GB |
| BIHOST*n* | 10 GB* |

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

### 5.1.3 Operating System Requirements for the Enterprise Deployment Topology

This section provides details about the operating system requirements.

The Oracle Fusion Middleware software products and components described in this guide are certified on various operating systems and platforms, which are listed in the *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 using the bash shell on Oracle Linux.

## 5.2 Reserving the Required IP Addresses for an Enterprise Deployment

This section lists the set of IP addresses that you must obtain and reserve before installation and configuration.

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 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 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.

**See Also:**

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 required for the Administration Server and each of the Managed Servers in a typical Oracle Business Intelligence enterprise deployment topology.

### 5.2.1 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, we reference *virtual* IP addresses, which can be re-assigned from one host to another, and *physical* IP addresses, which are assigned permanently to hardware host computer.

### 5.2.2 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 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 using the Whole Server Migration feature of Oracle WebLogic Server.

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

### 5.2.3 Physical and Virtual IP Addresses Required by the Enterprise Topology

This section describes the physical IP (IP) and virtual IP (VIP) addresses required for the Administration Server and each of the Managed Servers in a typical Oracle Business Intelligence 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 5-1.

You can assign any unique host name to the VIPs, but in this guide, we reference each VIP 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.

For more information, see Using the Enterprise Deployment Workbook

---

***Table 5-1 Summary of the Virtual IP Addresses Required for the Enterprise Deployment***

| Virtual IP | VIP Maps to... | Description |
|---|---|---|
| VIP1 | ADMINVHN | ADMINVHN is the virtual host name used as the listen address for the Administration Server and fails over with manual failover of the Administration Server. It is enabled on the node where the Administration Server process is running. |

***Table 5-1 (Cont.) Summary of the Virtual IP Addresses Required for the Enterprise Deployment***

| Virtual IP | VIP Maps to... | Description |
|---|---|---|
| VIP2 | BIHOST1VHN | BIHOST1VHN serves as the virtual host name that maps to the listen address for the WLS_BI1 Managed Server and fails over with Whole Server Migration of this server.<br>It is enabled in the node where the WLS_BI1 process is running (BIHOST1). |
| VIP3 | BIHOST2VHN | BIHOST2VHN serves as the virtual host name that maps to the listen address for the WLS_BI2 Managed Server and fails over with Whole Server Migration of this server.<br>It is enabled in the node where the WLS_BI2 process is running (BIHOST2). |

## 5.3 Identifying and Obtaining Software Downloads for an Enterprise Deployment

Before you begin installing and configuring the enterprise topology, you should locate and download the software distributions that you will need to implement the topology.

The following table lists the downloads you will need to obtain.

For general information about how to obtain Oracle Fusion Middleware software, see "Understanding and 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.

| Distribution | Description |
|---|---|
| Oracle Fusion Middleware 12*c* (12.2.1.0.0) Infrastructure | 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.<br>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 12*c* (12.2.1.0.0) | Download this distribution to install the Oracle HTTP Server software on the Web Tier. |
| Oracle Fusion Middleware 12*c* (12.2.1.0.0) Business Intelligence | Download this distribution to install the Oracle Business Intelligence software. |

# 6

# Preparing the Load Balancer and Firewalls for an Enterprise Deployment

This chapter describes how to configure your network for an enterprise deployment.

**See Also:**

Preparing for an Enterprise Deployment

Configuring Virtual Hosts on the Hardware Load Balancer
This section explains how to configure the hardware load balancer for an enterprise deployment.

Configuring the Firewalls and Ports for an Enterprise Deployment
This section lists the ports that muse be opened on the firewalls for an enterprise deployment.

## 6.1 Configuring Virtual Hosts on the Hardware Load Balancer

This section explains how to configure the hardware load balancer for an enterprise deployment.

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

**See Also:**

Overview of the Hardware Load Balancer Configuration

Typical Procedure for Configuring the Hardware Load Balancer

Summary of the Virtual Servers Required for an Enterprise Deployment

Additional Instructions for admin.example.com

### 6.1.1 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 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 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 in the request. For more information, see Configuring Oracle HTTP Server for Administration and Oracle Web Services Manager.

## 6.1.2 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. 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 or not a given host and service is available and assign it to the pool of servers 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 or not 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.

## 6.1.3 Summary of the Virtual Servers Required for an Enterprise Deployment

This section provides the details of the virtual servers required for an enterprise deployment.

The following table provides a list of the virtual servers you must define on the hardware load balancer for the Oracle Business Intelligence enterprise topology.

| Virtual Host | Server Pool | Protocol | SSL Termination? | External? |
|---|---|---|---|---|
| admin.example.com:80 | WEBHOST1.example.com:7777 WEBHOST2.example.com:7777 | HTTP | No | No |
| bi.example.com:443 | WEBHOST1.example.com:7777 WEBHOST2.example.com:7777 | HTTPS | Yes | Yes |
| biinternal.example.com:80 | WEBHOST1.example.com:7777 WEBHOST2.example.com:7777 | HTTP | No | No |

### 6.1.4 Additional Instructions for admin.example.com

This section provides the additional instructions required for the virtual server—admin.example.com.

When you configure this virtual server on the hardware load balancer:

- Enable address and port translation.
- Enable reset of connections when services or hosts are down.

## 6.2 Configuring the Firewalls and Ports for an Enterprise Deployment

This section lists the ports that muse be opened on the firewalls for an enterprise deployment.

As an administrator, you should be familiar with the port numbers used by the various Oracle Fusion Middleware products and services. You can then ensure that the same port number is not used by two services on a host, and you can make sure 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.

**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. |

| Type | Firewall | Port and Port Range | Protocol / Application | Inbound / Outbound | Other Considerations and Timeout Guidelines |
|---|---|---|---|---|---|
| Browser request | FW0 | 443 | HTTPS / Load Balancer | Inbound | Timeout depends on the size and type of HTML content. |
| Browser request | FW1 | 80 | HTTPS / 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 | HTTPS / 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 and HTTP, respectively | 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. |

| 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<br>t3 | Both | You should tune this timeout based on the type of access to the admin console (whether it is planned to use the Oracle WebLogic Server Administration Console from 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 | 8088<br>Range: 8000 - 8090 | | n/a | n/a |
| Oracle Unified Directory access | FW2 | 389<br>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. |

**Firewall Ports Specific to Oracle Business Intelligence Enterprise Deployments**

| Type | Firewall | Port and Port Range | Protocol / Application | Inbound / Outbound | Other Considerations and Timeout Guidelines |
|---|---|---|---|---|---|
| WSM-PM access | FW1 | 7010<br>Range: 7010 - 7999 | HTTP / WLS_WSM-PM*n* | Inbound | Set the timeout to 60 seconds. |
| BI Server access | FW1 | 9704 | HTTP / WLS_BI*n* | Inbound | Timeout varies based on the type of process model used for BI. |
| Communication between BI Cluster members | n/a | 9704 | TCP/IP Unicast | n/a | By default, this communication uses the same port as the server's listen address. |
| Database access for BI Server and BI Publisher JDBC data sources | FW1 | Listening port for client connections to the listener | SQL*Net | Inbound/ Outbound | Timeout depends on all database content and on the type of process model used for BI. |

# 7

# Preparing the File System for an Enterprise Deployment

Involves understanding the requirements for local and shared storage, as well as the terminology 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.

**See Also:**



## 7.1 Overview of Preparing the File System for an Enterprise Deployment

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

Use this chapter as a reference to help understand the directory variables 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.

## 7.2 Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment

This section provides reference to the shared storage recommendations when installing and configuring 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 7-1 lists the key sections you should review and how those concepts apply to an enterprise deployment.

***Table 7-1 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.<br>For the purposes of the enterprise deployment, Oracle recommends using redundant Oracle homes on separate storage volumes.<br>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.<br>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. |

**Note:** Oracle Business Intelligence has an additional shared storage requirement for setting the location of specific Business Intelligence metadata. For more information, see Configuring the Singleton Data Directory (SDD).

## 7.3 Understanding the Recommended Directory Structure for an Enterprise Deployment

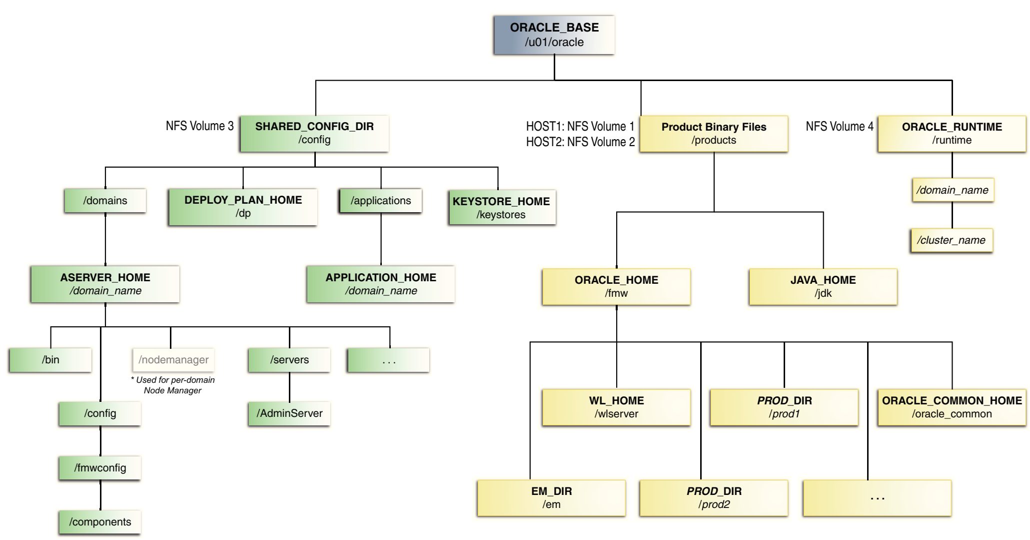
The diagrams in this section show the recommended directory structure for a typical Oracle Fusion Middleware enterprise deployment.

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

- Figure 7-1 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 7-2 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 7-3 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.

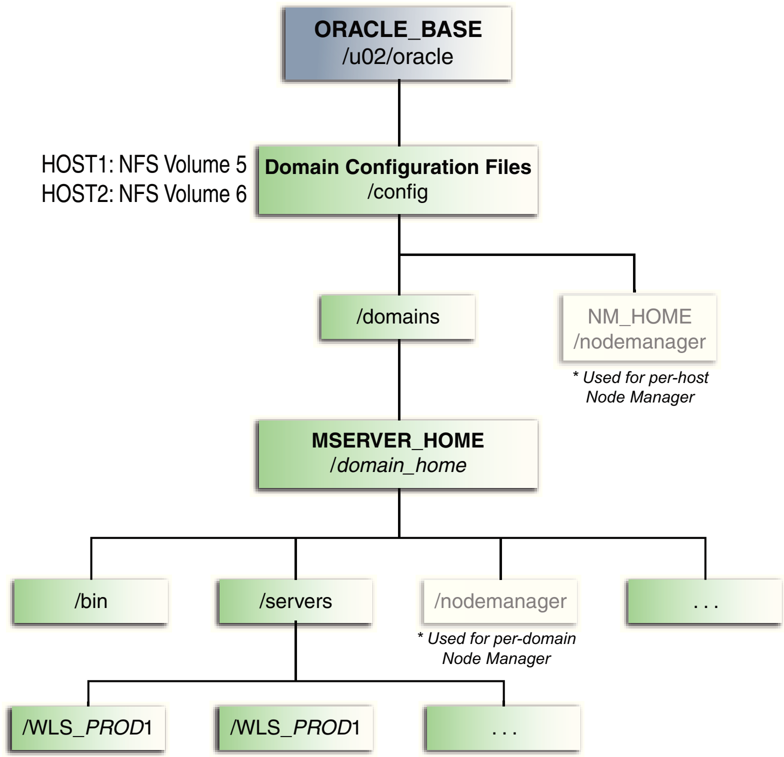
Where applicable, the diagrams also include the standard variables used to reference the directory locations in the installation and configuration procedures in this guide.

***Figure 7-1 Recommended Shared Storage Directory Structure for an Enterprise Deployment***



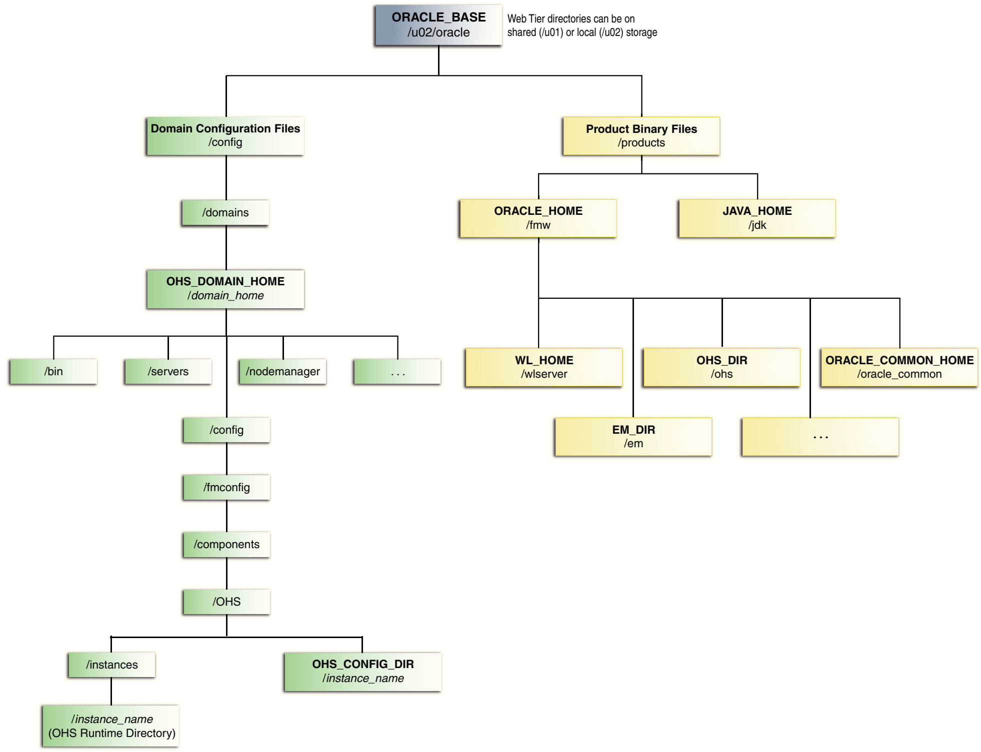

* For more information, see About the Node Manager Configuration in a Typical Enterprise Deployment.

***Figure 7-2 Recommended Local Storage Directory Structure for an Application Tier Host Computer in an Enterprise Deployment***




* For more information, see About the Node Manager Configuration in a Typical Enterprise Deployment.

***Figure 7-3 Recommended Local Storage Directory Structure for a Web Tier Host Computer in an Enterprise Deployment***




# 7.4 File System and Directory Variables Used in This Guide

This section lists and describes the file system and directory variables used throughout this guide.

Table 7-2 lists the file system directories and the directory variables used to reference the directories on the Application tier. Table 7-3 lists the file system directories and variables used to reference the directories on the Web tier.

For additional information about mounting these directories when you are using 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 using 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 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.

For more information, see Using the Enterprise Deployment Workbook..

---

***Table 7-2 Sample Values for Key Directory Variables on the Application Tier***

| Directory Variable | Description | Sample Value on the Application Tier |
|---|---|---|
| *ORACLE_BASE* | The base directory, under which Oracle products are installed. | `/u01/oracle` |
| *ORACLE_HOME* | The read-only location for the product binaries. For the application tier host computers, it is stored on shared disk.<br>The Oracle home is created when you install the Oracle Fusion Middleware Infrastructure software.<br>You can then install additional Oracle Fusion Middleware products into the same Oracle home. | `/u01/oracle/products/fmw` |
| *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. | `/u01/oracle/products/fmw/oracle_common` |
| *WL_HOME* | The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored. | `/u01/oracle/products/fmw/wlserver` |
| *PROD_DIR* | Individual product directories for each Oracle Fusion Middleware product you install. | `/u01/oracle/products/fmw/`*`prod_dir`*<br>The product can be `soa`, `wcc`, `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. | `/u01/oracle/products/fmw/em` |
| *JAVA_HOME* | The location where you install the supported Java Development Kit (JDK). | `/u01/oracle/products/jdk` |
| *SHARED_CONFIG_DIR* | The shared parent directory for shared environment configuration files, including domain configuration, keystores, runtime artifacts, and application deployments | `/u01/oracle/config` |
| *ASERVER_HOME* | The Administration Server domain home, which is installed on shared disk. | `/u01/oracle/config/domains/`*`domain_name`*<br>In this example, replace domain_name with the name of the WebLogic Server domain. |
| *MSERVER_HOME* | The Managed Server domain home, which is created via the unpack command on the local disk of each application tier host. | `/u02/oracle/config/domains/`*`domain_name`* |

***Table 7-2 (Cont.) Sample Values for Key Directory Variables on the Application Tier***

| Directory Variable | Description | Sample Value on the Application Tier |
|---|---|---|
| *APPLICATION_HOME* | The Application home directory, which is installed on shared disk, so the directory is accessible by all the application tier host computers. | `/u01/oracle/config/applications/domain_name` |
| *ORACLE_RUNTIME* | This directory contains the Oracle runtime artifacts, such as the JMS logs and TLogs.<br>Typically, you mount this directory as a separate shared file system, which is accessible by all hosts in the domain.<br>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.<br>For example:<br>`ORACLE_RUNTIME/cluster_name/jms` | `/u01/oracle/runtime/` |
| *NM_HOME* | The directory used by the Per Machine Node Manager start script and configuration files.<br>**Note:** This directory is necessary only if you are using a Per Machine Node Manager configuration.<br>For more information, see About the Node Manager Configuration in a Typical Enterprise Deployment. | `/u02/oracle/config/node_manager` |
| *DEPLOY_PLAN_HOME* | The deployment plan directory, which is used as the default location for application deployment plans.<br>**Note:** This directory is required only when you are deploying custom applications to the application tier. | `/u01/oracle/config/dp` |

**Table 7-2 (Cont.) Sample Values for Key Directory Variables on the Application Tier**

| Directory Variable | Description | Sample Value on the Application Tier |
|---|---|---|
| *KEYSTORE_HOME* | The shared location for custom certificates and keystores. | `/u01/oracle/config/keystores` |

**Table 7-3 Sample Values for Key Directory Variables on the Web Tier**

| Directory Variable | Description | Sample Value on the Web Tier |
|---|---|---|
| *OHS_ORACLE_HOME* | The read-only location for the Oracle HTTP Server product binaries. For the Web tier host computers, this directory is stored on local disk.<br>The Oracle home is created when you install the Oracle HTTP Server software. | `/u02/oracle/products/fmw` |
| *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/oracle/products/fmw/oracle_common` |
| *WL_HOME* | The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored. | `/u02/oracle/products/fmw/wlserver` |
| *PROD_DIR* | Individual product directories for each Oracle Fusion Middleware product you install. | `/u02/oracle/products/fmw/ohs` |
| *JAVA_HOME* | The location where you install the supported Java Development Kit (JDK). | `/u02/oracle/products/jdk` |
| *OHS_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/oracle/config/domains/`*`domain_name`* |
| *OHS_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.<br>Note this directory is also referred to as the OHS Staging Directory. Changes made here are later propagated to the OHS Runtime Directory.<br>For more information, see “Staging and Run-time Configuration Directories” in the *Administrator's Guide for Oracle HTTP Server*. | `/u02/oracle/config/domains/`*`domain_name`*`/config/fmwconfig/components/OHS/`*`instance_name`* |

## 7.5 About Creating and Mounting the Directories for an Enterprise Deployment

This section lists the best practices to be followed when creating or mounting the top-level directories in an enterprise deployment.

When creating or mounting the top-level directories, note the following best practices:

- 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 BIHOST2. Be sure the directory path to the binaries on BIHOST2 is identical to the directory path on BIHOST1.

  For example:

  ```
  /u01/oracle/products/fmw/
  ```

  For more information, see Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

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

  The Web tier installation is typically performed on local storage to the WEBHOST nodes. When using shared storage, you can install the Oracle Web tier binaries (and create the Oracle HTTP Server instances) on 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 (BIHOST1 and BIHOST2), use the same directory path on both computers.

  For example:

  ```
  /u02/oracle/products/fmw/
  ```

## 7.6 Summary of the Shared Storage Volumes in an Enterprise Deployment

This section provides a summary of the shared storage volumes required for an enterprise deployment.

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

For more information, see Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment.

| Volume in Shared Storage | Mounted to Host | Mount Directories | Description and Purpose |
|---|---|---|---|
| NFS Volume 1 | BIHOST1 | `/u01/oracle/products/` | Local storage for the product binaries to be used by BIHOST1; this is where the Oracle home directory and product directories are installed. |

| Volume in Shared Storage | Mounted to Host | Mount Directories | Description and Purpose |
|---|---|---|---|
| NFS Volume 2 | BIHOST2 | `/u01/oracle/products/` | Local storage for the product binaries to be used by BIHOST2; this is where the Oracle home directory and product directories are installed. |
| NFS Volume 3 | BIHOST1 BIHOST2 | `/u01/oracle/config/` | Administration Server domain configuration, mounted to all hosts; used initially by BIHOST1, but can be failed over to any host. |
| NFS Volume 4 | BIHOST1 BIHOST2 | `/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. |
| NFS Volume 5 | BIHOST1 | `/u02/oracle/config/` | Local storage for the Managed Server domain directory to be used by BIHOST1, if the private Managed Server domain directory resides on shared storage. |
| NFS Volume 6 | BIHOST2 | `/u02/oracle/config/` | Local storage for the Managed Server domain directory to be used by BIHOST2, if the private Managed Server domain directory resides on shared storage. |
| NFS Volume 7 | WEBHOST1 | `/u02/oracle/` | Local storage for the Oracle HTTP Server software binaries (Oracle home) and domain configuration files used by WEBHOST1, if the private Managed Server domain directory resides on shared storage. |

| Volume in Shared Storage | Mounted to Host | Mount Directories | Description and Purpose |
|---|---|---|---|
| NFS Volume 8 | WEBHOST2 | `/u02/oracle/` | Local storage for the Oracle HTTP Server software binaries (Oracle home) and domain configuration files used by WEBHOST2, if the private Managed Server domain directory resides on shared storage. |

# 8

# Preparing the Host Computers for an Enterprise Deployment

It explains how to mount the required shared storage systems to the host and how to enable the required virtual IP addresses on each host.

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

**See Also:**



## 8.1 Verifying the Minimum Hardware Requirements for Each Host

This section provides information about the minimum hardware requirements for each host.

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.

If you are deploying to a virtual server environment, such as Oracle Exalogic, ensure that each of the virtual servers meets the minimum requirements.

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 `/tmp`.

# 8.2 Verifying Linux Operating System Requirements

Review this section for typical Linux operating system settings for an enterprise deployment.

To ensure the host computers meet the minimum operating system requirements, be sure 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.

**See Also:**

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

## 8.2.1 Setting Linux Kernel Parameters

The kernel-parameter and shell-limit values shown below 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 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 are deploying a database onto the host, you might need to modify additional kernel parameters. Refer to the 12c (12.2.1) *Oracle Grid Infrastructure Installation Guide* for your platform.

***Table 8-1 UNIX Kernel Parameters***

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

To set these parameters:

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

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

## 8.2.2 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.

**See Also:**

Viewing the Number of Currently Open Files

Setting the Operating System Open File and Processes Limits

### 8.2.2.1 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
```

### 8.2.2.2 Setting the Operating System Open File and Processes Limits

To change the Open File Limit values:

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

   ```
   /etc/security/limits.conf
   ```

2. Add the following lines to the `limits.conf` 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 `limits.conf` file.

4. Reboot the host computer.

### 8.2.3 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
```

## 8.3 Configuring Operating System Users and Groups

The lists in this section show the users and groups to define on each of the computers that will host the enterprise deployment.

**Groups**

You must create the following groups on each node.

- `oinstall`
- `dba`

**Users**

You must create the following user on each node.

- `nobody`–An unprivileged user.
- `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.

---

## 8.4 Enabling Unicode Support

This section provides information about enabling Unicode support.

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 Business Intelligence technologies, for example, are based on Unicode.

If the operating system is configured to use a non-UTF-8 encoding, Oracle Business Intelligence 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.

## 8.5 Mounting the Required Shared File Systems on Each Host

This section provides information about mounting the required shared file systems on each host.

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 you have procured for the depoyment.

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 using NFS storage.

You must create and mount shared storage locations so that BIHOST1 and BIHOST2 can see the same location if it is a binary installation in two separate volumes.

For more information, 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 execute the files, but they do not have write privileges.

For more information about installation and configuration 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 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.

1. Create the mount directories on BIHOST1, as described in Summary of the Shared Storage Volumes in an Enterprise Deployment, and then mount the shared storage. For example:

   ```
   mount -t nfs nasfiler:VOL1/oracle/products/ /u01/oracle/products/
   ```

2. Repeat the procedure on BIHOST2 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 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 BIHOST1. 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 for the correct options for your environment.

---

## 8.6 Enabling the Required Virtual IP Addresses on Each Host

This section provides instruction to enable the required virtual IP addresses on each host.

To prepare each host for the enterprise deployment, you must enable the virtual IP (VIP) addresses 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).

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

```
/sbin/ifconfig interface:index IPAddress netmask netmask
/sbin/arping -q -U -c 3 -I interface IPAddress
```

where *`interface`* is `eth0`, or `eth1`, and *`index`* is `0`, `1`, or `2`.

For example:

```
/sbin/ifconfig eth0:1 100.200.140.206 netmask 255.255.255.0
```

Enable your network to register the new location of the virtual IP address:

```
/sbin/arping -q -U -c 3 -I eth0 100.200.140.206
```

Validate that the address is available by using the `ping` command from another node, for example:

```
/bin/ping 100.200.140.206
```

# 9 Preparing the Database for an Enterprise Deployment

This chapter describes procedures for preparing your database for an Oracle Business Intelligence enterprise deployment.

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

**See Also:**



## 9.1 Overview of Preparing the Database for an Enterprise Deployment

This section provides information about how 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 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.

## 9.2 About Database Requirements

Check that the database meets the requirements described in these sections.

**See Also:**

Supported Database Versions

Additional Database Software Requirements

### 9.2.1 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 using 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*.

### 9.2.2 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. We refer to these hosts as DBHOST1 and DBHOST2.

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

- **Oracle Clusterware**

  For more information, see the *Oracle Grid Infrastructure Installation Guide for Linux*.

- **Oracle Real Application Clusters**

  For more information, see the *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.

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

The clocks of the database instances being used by servers in a Fusion Middleware cluster that is configured with server migration must be in sync.

- **Automatic Storage Management** (optional)

  For more information, see the *Oracle Automatic Storage Management Administrator's Guide*.

## 9.3 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.

---

**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.

---

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

Run-time 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 using the `srvctl` utility.

To create and modify a database service:

1. Log in to SQL*Plus and create the service:

```
sqlplus "sys/password as sysdba"

SQL> EXECUTE DBMS_SERVICE.CREATE_SERVICE
(SERVICE_NAME => 'biedg.example.com',
NETWORK_NAME => 'biedg.example.com'
);
```

---

**Note:**

For the Service Name of the Oracle RAC database, use lowercase letters, followed by the domain name. For example:

`biedg.example.com`

---

**Note:**

Enter the `EXECUTE DBMS_SERVICE` command shown on a single line.

For more information about the `DBMS_SERVICE` package, see *Oracle Database PL/SQL Packages and Types Reference*.

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

```
srvctl add service -d bidb -s biedg.example.com -r bidb1,bidb2
```

3. Start the service:

```
srvctl start service -d bidb -s biedg.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*.

4. Modify the service so it uses the Load Balancing Advisory and the appropriate service-level goals for run-time connection load balancing.

   More specifically, 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”

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

Beginning with Oracle Database 11g Release 1, Oracle introduced SecureFiles, a new LOB storage architecture. Oracle recommends using SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas.

For more information, 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 explicitely specify BasicFiles, which means that Oracle Fusion Middleware LOBs will default 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`: create all (new) LOBs as SecureFiles

- `ALWAYS`: try to create LOBs as SecureFiles, but fall 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 creating 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.

## 9.5 About Database Backup Strategies

This section provides brief information about the necessity of database backup strategies.

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 using operating system tools or RMAN for this purpose.

Oracle recommends using 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 using operating system tools such as tar.

# Part III

# Configuring the Enterprise Deployment

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

**See Also:**


This chapter describes the steps to scale out your initial Oracle Business Intelligence domain to BIHOST2.

# 10

# Creating the Initial BI Domain for an Enterprise Deployment

This chapter describes how to install and configure an Oracle Business Intelligence (BI) domain, which can be used as the starting point for an enterprise deployment.

This chapter contains information on variables used when creating the BI domain, creating database schemas and configuring the BI domain.

**See Also:**

Configuring the Enterprise Deployment

global cache so that cache seeding and purging events can be shared by all Oracle BI servers participating in a cluster.

Verifying Oracle Business Intelligence URLs on BIHOST1
After starting the components in the domain on BIHOST1, access these URLs to verify the configuration of Oracle Business Intelligence.

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.

Backing Up the Oracle Business Intelligence Configuration
It is an Oracle best practices recommendation to create a backup after successfully configuring a domain or at another logical point. Create a backup after verifying 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.

## 10.1 Variables Used When Creating the BI Domain

As you perform the tasks in this chapter, you will be referencing the directory variables listed in this section.

The directory variables are defined in File System and Directory Variables Used in This Guide.

- ORACLE_HOME
- ASERVER_HOME
- 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
- BIHOST1VHN1
- BIHOST1
- SCAN Address for the Oracle RAC Database (DB-SCAN.example.com)

## 10.2 Understanding the Initial BI Domain

Before you being creating the initial Business Intelligence (BI) domain, be sure to review the following key concepts.

**See Also:**

About the Infrastructure Distribution

Characteristics of the Initial BI Domain

### 10.2.1 About the Infrastructure Distribution

You create the initial Business Intelligence domain for an enterprise deployment, using the Oracle Fusion Middleware Infrastructure distribution. This distribution contains both the Oracle WebLogic Server software and the Oracle JRF software in one distribution.

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 required to support the Oracle Fusion Middleware products.

For more information, see "Understanding Oracle Fusion Middleware Infrastructure" in *Understanding Oracle Fusion Middleware*.

### 10.2.2 Characteristics of the Initial BI Domain

Review these key characteristics of the initial BI domain. By reviewing and understanding these characteristics, you can better understand the purpose and context of the procedures used to configure the domain.

Many of these characteristics are described in more detail in Understanding a Typical Enterprise Deployment.

***Table 10-1 Characteristics of the Initial BI domain***

| 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 Per Domain Node Manager and separate Node Manager processes for the Administration Server and Managed Servers on each host. | 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 |

## 10.3 Installing the Oracle Fusion Middleware Infrastructure in Preparation for an Enterprise Deployment

Use this section to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for an enterprise deployment.

**See Also:**

Installing a Supported JDK

Starting the Infrastructure Installer on BIHOST1

Navigating the Infrastructure Installation Screens

Checking the Directory Structure

## 10.3.1 Installing a Supported JDK

Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system. See the following sections for more information:

**See Also:**

Locating and Downloading the JDK Software

Installing the JDK Software

### 10.3.1.1 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.

### 10.3.1.2 Installing the JDK Software

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.
- 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.

For more information about the recommended location for the JDK software, see the Understanding the Recommended Directory Structure for an Enterprise Deployment.

The following example describes how to install a recent version of JDK 1.8:

1. Change directory to the location where you downloaded the JDK archive file.

2. Unpack the archive into the JDK home directory, and then run these commands:

   ```
   cd download_dir
   tar -xzvf jdk-8u51-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_51 /u01/oracle/products/jdk
   ```

   For more information, 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 -version
     java version "1.8.0_51"
     Java(TM) SE Runtime Environment (build 1.8.0_51-b16)
     Java HotSpot(TM) 64-Bit Server VM (build 25.51-b03, mixed mode)
   ```

## 10.3.2 Starting the Infrastructure Installer on BIHOST1

To start the installation program, perform the following steps.

1. Log in to BIHOST1.

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.

     Note that 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_12.2.1.0.0_infrastructure.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.

## 10.3.3 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.

| Screen | Description |
|---|---|
| Installation Inventory Setup | On UNIX operating systems, this screen will appear if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location.<br>For more information about the central inventory, see "Understanding the Oracle Central Inventory" in *Installing Software with the Oracle Universal Installer*. |
| 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.<br>For the purposes of an enterprise deployment, enter the value of the *ORACLE_HOME* variable listed in Table 7-2. |
| Installation Type | Use this screen to select the type of installation and consequently, the products and feature sets you want to install.<br>For this topology, select **Fusion Middleware Infrastructure**.<br>**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 necessary requirements.<br>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.<br>If you do not have one and are sure you want to skip this step, clear the check box and verify your selection in the follow-up dialog box. |
| Auto Updates - Patch Selection | This screen appears if both of the following statements are true:<br>• You searched for available patches earlier in the installation session, using the Auto Updates screen.<br>• The Auto Updates feature located one or more application patches that must be applied to the Oracle home you are creating in this installation session.<br>This screen lists the patches that were found by the Auto Updates feature. Select one or more patches and click **Next** to apply the selected patches to the Oracle home. |

| Screen | Description |
|---|---|
| Installation Summary | Use this screen to verify the installation options 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.<br><br>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. |

## 10.3.4 Checking the Directory Structure

After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the following directory and sub-directories. The contents of your installation vary based on the options you selected during the installation.

To check the directory structure:

1. Change directory to the *ORACLE_HOME*directory.
2. Enter the following command:

   ```
   ls -l
   ```

   The directory structure on your system should match the structure shown in the following example:

   ```
   /u01/oracle/products/fmw/

   cfgtoollogs
   coherence
   em
   install
   inventory
   OPatch
   oracle_common
   oraInst.loc
   oui
   root.sh
   wlserver
   ```

   For more information about the directory structure after the installation complete, see "What are the Key Oracle Fusion Middleware Directories?" in *Understanding Oracle Fusion Middleware*.

# 10.4 Installing Oracle Business Intelligence in Preparation for an Enterprise Deployment

Use this section to install the Oracle Business Intelligence software in preparation for configuring a new domain for an enterprise deployment.

**See Also:**

Starting the Installation Program

Navigating the Installation Screens

Checking the Directory Structure

## 10.4.1 Starting the Installation Program

Use these steps to start the BI Installer.

1. Log in to BIHOST1.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the executable, as shown in the example below.

```
./bi_platform-12.2.1.0.0_linux64.bin
```

For more information about the actual file names for 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.

## 10.4.2 Navigating the Installation Screens

The installation program displays a series of screens, in the order listed in Table 10-2.

If you need additional help with any of the installation screens, click the screen name.

***Table 10-2 Oracle Business Intelligence Install Screens***

| Screen | Description |
|---|---|
| Installation Inventory Setup | On UNIX operating systems, this screen will appear if this is the first time you are installing any Oracle product on this host. Specify the location where you want to create your central inventory. Make sure that the operating system group name selected on this screen has write permissions to the central inventory location.<br><br>For more information about the central inventory, see "Understanding the Oracle Central Inventory" in *Installing Software with the Oracle Universal Installer*. |
| Welcome | This screen introduces you to the product installer. |

***Table 10-2 (Cont.) Oracle Business Intelligence Install Screens***

| Screen | Description |
|---|---|
| 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.<br>For the purposes of an enterprise deployment, enter the value of the *ORACLE_HOME* variable listed in Table 7-2. |
| Installation Type | Use this screen to select the type of installation and consequently, the products and feature sets you want to install.<br>For this topology, select **BI Platform Distribution with Samples**. |
| Prerequisite Checks | This screen verifies that your system meets the minimum necessary requirements.<br>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). |
| Auto Updates - Patch Selection | This screen appears if the both of the following statements are true:<br>• You searched for available patches earlier in the installation session, using the Auto Updates screen.<br>• The Auto Updates feature located one or more application patches that must be applied to the Oracle home you are creating in this installation session.<br>This screen lists the patches that were found by the Auto Updates feature. Select one or more patches and click **Next** to apply the selected patches to the Oracle home. |
| Installation Summary | Use this screen to verify the installation options 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.<br>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. |

### 10.4.3 Checking the Directory Structure

The contents of your installation vary based on the options you selected during the installation.

After you install Oracle Business Intelligence, you should see the following directory and sub-directories.

```
/u01/oracle/products/fmw1221/bi

bi-epm-registry
bifoundation
bin
clients
common
endpointmanager
file_templates
jlib
lib
migration-tool
modules
nls
oracore
plugins
products
schema
vcredist_x64.exe
vcredist_x86.exe
xsd
```

## 10.5 Creating the Database Schemas

Before you can configure a BI domain, you must install the schemas listed in this section on 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)
- WebLogic Runtime Services (WLS_RUNTIME)
- Common Infrastructure Services (STB)
- Business Intelligence Platform (BIPLATFORM)

You 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*.

**See Also:**

Installing and Configuring a Certified Database

Starting the Repository Creation Utility (RCU)

Navigating the RCU Screens to Create the Schemas

### 10.5.1 Installing and Configuring a Certified Database

Make sure you have installed and configured a certified database, and that the database is up and running.

For more information, see the following resources:

- Preparing the Database for an Enterprise Deployment, which includes information about creating database services, using SecureFiles for Large Objects (LOBs), and other topics important in an enterprise deployment.
- Understanding Database Requirements for an Oracle Fusion Middleware Installation in *Planning an Installation of Oracle Fusion Middleware*.

### 10.5.2 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.

   For more information, see File System and Directory Variables Used in This Guide.

2. Navigate to the following directory on BIHOST1:

   ```
   ORACLE_HOME/oracle_common/bin
   ```

3. Start RCU:

   ```
   ./rcu
   ```

### 10.5.3 Navigating the RCU Screens to Create the Schemas

Follow the instructions in this section to create the schemas for the Oracle Business Intelligence 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*.

**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 Credentials**

On the Database Connection Details screen, provide the database connection details for RCU to connect to your database.

In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.

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 the purposes of this guide, use the prefix `FMW1221`.

   **Tip:**

   Make a note of the custom prefix you choose to enter here; you will need this later during the domain creation process.

2. Select **AS Common Schemas**.

   When you select **AS Common Schemas**, all of the schemas in this section are automatically selected.

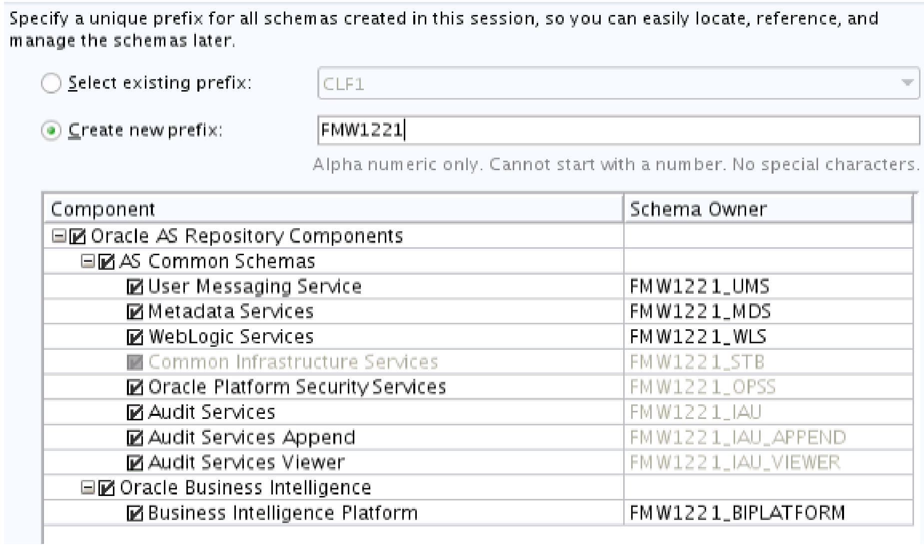
   A schema called **Common Infrastructure Services** is also automatically created; this schema is grayed out and cannot be selected or deselected. This schema (the STB schema) enables you to retrieve information from RCU during domain configuration. For more information, see "Understanding the Service Table Schema" in *Creating Schemas with the Repository Creation Utility*.

3. Select **Business Intelligence Platform**.

   **Tip:**

   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*.




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.

> **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 Completing Schema Creation**

Navigate through the remainder of the RCU screens to complete schema creation.

For the purposes of this guide, 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.

For more information about RCU and its features and concepts, see *Creating Schemas with the Repository Creation Utility*.

When you reach the Completion Summary screen, click **Close** to dismiss RCU.

## 10.6 Configuring the BI Domain

This section provides instructions for creating a WebLogic domain using the 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*.

The following tasks are covered in this section.

**See Also:**

Starting the Configuration Wizard

Navigating the Configuration Wizard Screens to Configure the BI Domain

### 10.6.1 Starting the Configuration Wizard

To begin domain configuration, run the following command in the Oracle Fusion Middleware Oracle home.

```
ORACLE_HOME/oracle_common/common/bin/config.sh
```

### 10.6.2 Navigating the Configuration Wizard Screens to Configure the BI Domain

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 can be found in "Configuration Type" in *Creating WebLogic Domains Using the Configuration Wizard*.

**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 BIEE Suite – 12.2.1.0 [bi]**

  Selecting this template automatically selects the following dependencies:

  – Oracle MapViewer –12.2.1 [oracle_common]

  – Oracle Enterprise Manager – 12.2.1 [em]

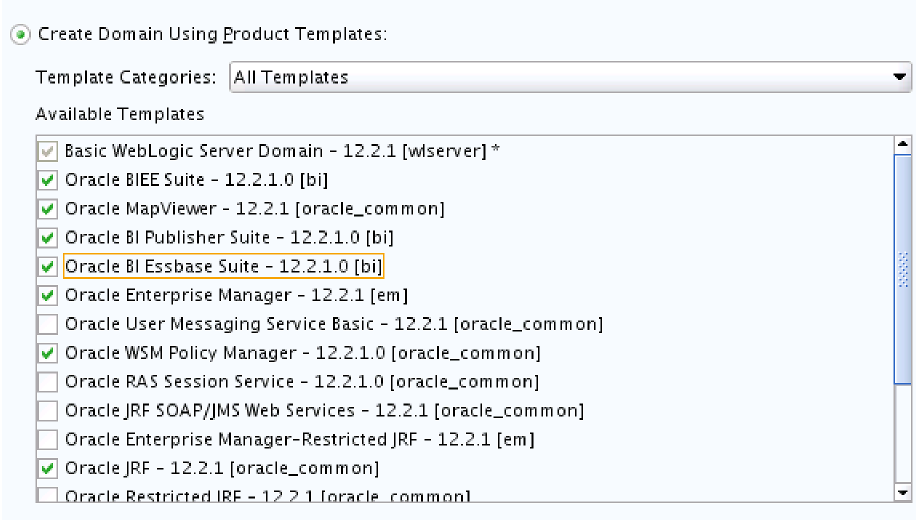
  – Oracle WSM Policy Manager – 12.2.1.0 [oracle_common]

  – Oracle JRF - 12.2.1 [oracle_common]

  – WebLogic Coherence Cluster Extension - 12.2.1 [wlserver]

- **Oracle BI Publisher Suite – 12.2.1.0 [bi]**
- **Oracle BI Essbase Suite – 12.2.1 [bi]**

In addition, the **Basic WebLogic Server Domain – 12.2.1 [wlserver]** template should already be selected and grayed out.




**Tip:**
More information about the options on this screen can be found in Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

**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*.

**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.

**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. For more information, see Creating the boot.properties File.

**Task 6 Specifying the Database Configuration Type**

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.

---

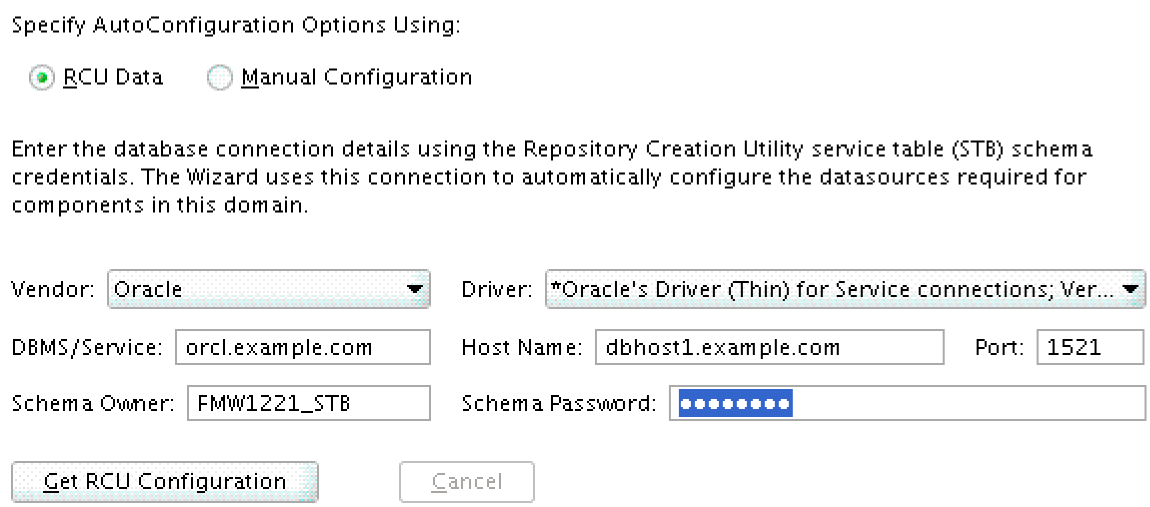
**Note:**
If you choose to select **Manual Configuration** on this screen, you will have to manually fill in the parameters for your schema on the JDBC Component Schema screen.

---

After selecting **RCU Data**, fill in the fields as shown in the following table. Refer to Figure 10-1 for a partial screen shot of a sample Database Configuration Type screen.

| Field | Description |
|---|---|
| DBMS/Service | Enter the service name for the Oracle RAC database where you will install the product schemas. For example:<br>`orcl.example.com`<br>Be sure this is 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.<br>For more information, see Preparing the Database for an Enterprise Deployment. |
| Host Name | Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the *Enterprise Deployment Workbook*. |
| Port | Enter the port number on which the database listens. For example, `1521`. |
| Schema Owner<br>Schema Password | Enter the user name and password for connecting to the database's Service Table schema.<br>This is the schema user name and password that was specified for the Service Table component on the Schema Passwords screen in RCU.<br>The default user name is *`prefix`*`_STB`, where *`prefix`* is the custom prefix that you defined in RCU. |

***Figure 10-1 Setting the Database Configuration Type for an Enterprise Deployment***




Click **Get RCU Configuration** when you are finished specifying the database connection information. The following output in the Connection Result Log indicates that the operation 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 check box in the first header row of the schema table.

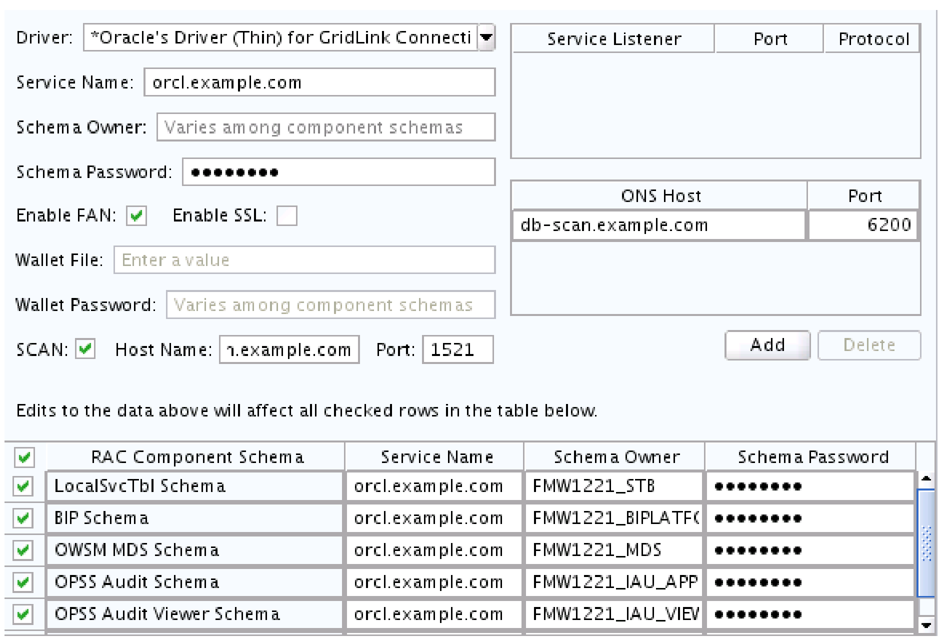
2. Click **Convert to GridLink** and click **Next**.

**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 Table 10-3 and in Figure 10-2.

| Element | Description and Recommended Value |
|---|---|
| SCAN, Host Name, and Port | Select the **SCAN** check box.<br>In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database.<br>In the **Port** field, enter the SCAN listening port for the database (for example, `1521`) |
| ONS Host and Port | In the **ONS Port** field, enter the SCAN address for the Oracle RAC database.<br>In the Port field, enter the ONS Remote port (typically, `6200`). |
| Enable Fan | Select the **Enable Fan** check box to receive and process FAN events, |

***Figure 10-2 Sample Values for the GridLink Oracle RAC Component Schema Screen***




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.

**Task 9 Testing the JDBC Connections**

Use the JDBC Component Schema Test screen to test the data source connections you have just configured.

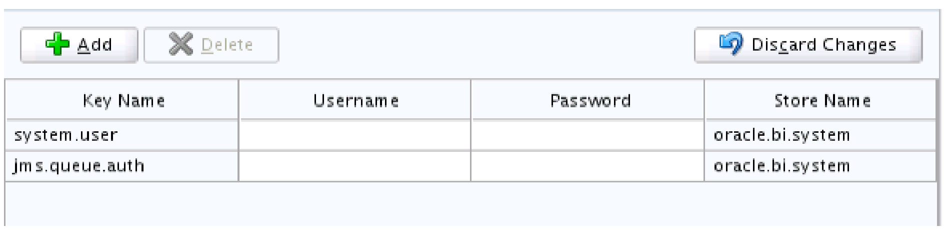
A green check mark in the Status column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **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*

**Task 10 Specifying Credentials**

Enter a unique user name and password for the Business Intelligence `system.user` account. Note that the `system.user` account is not an actual user. It is used for internal authentication between the different Business Intelligence components. You must provide a unique, random user name and password that are not used by an actual system user to log in and use BI applications with.

Enter a user name and password for the `jms.queue.auth` user account. This user must be a user in the WebLogic Administrator group.




**Task 11 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.

- **Managed Servers, Clusters and Coherence**

  This is required to configure the Managed Server and cluster, and also for configuring the machine and targeting the Managed Server to the machine.

- **JMS File Store**

  This is required to configure the appropriate shared storage for JMS persistent stores.

---

**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.

---

**Task 12 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. Leave the other fields at their default values.

   In particular, be sure that no server groups are assigned to the Administration Server.

**Task 13 Configuring Node Manager**
Select **Per Domain Default Location** as the Node Manager type, then specify the Node Manager credentials you will use to connect to the Node Manager.

> **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*.

**Task 14 Configuring the Managed Server**
On the Managed Servers screen, a new Managed Server for Oracle Business Intelligence appears in the list of servers. This server was created automatically by the **Oracle BIEE Suite** configuration template you selected on the Templates screen. Perform the following tasks to modify the default Oracle Business Intelligence Managed Server (`bi_server1`).

1. Rename the default Managed Server to `WLS_BI1`.

**Tip:**

The server name recommended here will be used throughout this document; if you choose a different name, be sure to replace it as needed.

2. Use the information in the following table to fill in the rest of the columns for the Oracle Business Intelligence 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*.

| Server Name | Listen Address | Listen Port | Enable SSL | SSL Listen Port | Server Groups |
|---|---|---|---|---|---|
| WLS_BI1 | BIHOST1 | 7003 | No | Disabled | BISUITE_MAN-SVR |

**Task 15 Configuring a Cluster**
In this task, you create a cluster to which you can target the Oracle BI software.

You will also set the **Frontend Host** property for the cluster, which ensures that, when necessary, WebLogic Server will redirect Web services callbacks and other redirects to `bi.example.com` on the load balancer rather than the address in the HOST header of each request.

For more information about the `bi.example.com` virtual server address, see Configuring Virtual Hosts on the Hardware Load Balancer.

On the Clusters screen, a new cluster (`bi_cluster`) for Oracle Business Intelligence appears in the list of clusters. Perform the following tasks to modify the default Oracle Business Intelligence cluster:

1. Specify `bi.example.com` in the **Frontend Host** field.

2. Specify `80` as the **Frontend HTTP Port** and `443` as the **Frontend HTTPS** port.

**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*.

**Task 16 Assigning the Managed Server to the Cluster**
Use the Assign Servers to Clusters screen to assign `WLS_BI1` to the new cluster `bi_cluster`:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, `bi_cluster`.

2. In the Servers pane, assign `WLS_BI1` to `bi_cluster` by doing one of the following:

   - Click once on `WLS_BI1` 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 `WLS_BI1` to move it beneath the selected cluster in the clusters pane.

   **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 17 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, refer to "Oracle Coherence" in *Oracle Fusion Middleware Licensing Information*.

**Task 18 Creating Machines**

Use the Machines screen to create a new machine in the domain. A machine is required in order for the Node Manager to be able to start and stop the servers.

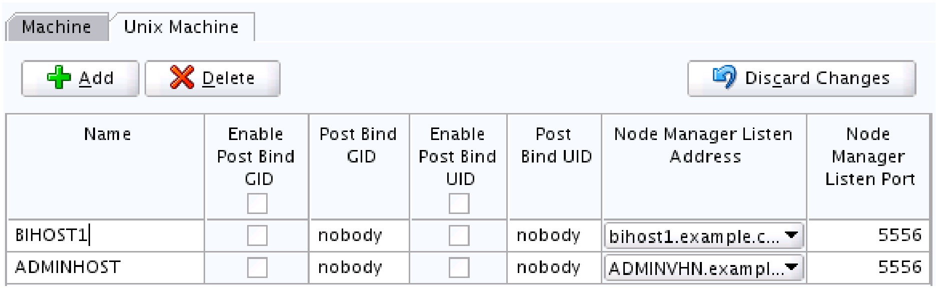
1. Select the **Unix Machine** tab.
2. Click the **Add** button to create the new UNIX machine.

   Use the values in the following table 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 |
|---|---|---|
| BIHOST1 | The value of the BIHOST1 host name variable. For example, `BIHOST1.example.com`. | 5556 |
| ADMINHOST | Enter the value of the ADMINVHN variable. | 5556 |

***Figure 10-3 Example Values for the Configuration Wizard Unix Machines Screen***




**Tip:**
More information about the options on this screen can be found in Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

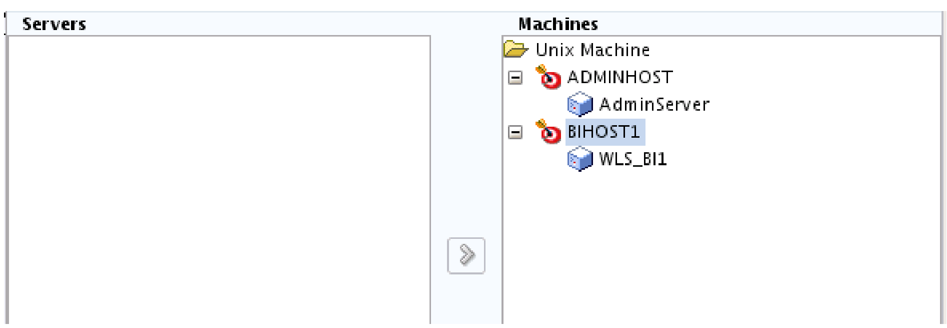
### Task 19 Assigning Servers to Machines

Use the Assign Servers to Machines screen to assign the Administration Server and the Oracle BI EE Suite Managed Server to the appropriate machine.

The Assign Servers to Machines screen is similar to the Assign Managed Servers to Clusters screen. Select the target machine in the Machines column, select the Managed Server in the left column, and click the right arrow to assign the server to the appropriate machine.

Assign the servers as follows:

- Assign the AdminServer to the ADMINHOST machine.
- Assign the WLS_BI1 Managed Server to the BIHOST1 machine.




**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*.

### Task 20 Configuring the JMS File Store

When you configure a domain using the Oracle WSM Policy Manager configuration template, you should select the proper location of the Metadata Services (MDS) JMS File Store, especially when you are configuring an enterprise deployment.

In the JMS File Stores screen, assign the following directory for each of the BI Persistence stores:

`ASERVER_HOME/bi_cluster`

In this example, replace ASERVER_HOME with the actual value of the ASERVER_HOME variable, as defined in File System and Directory Variables Used in This Guide. Replace `bi_cluster` with the name you assigned to the Oracle BI cluster.

#### Task 21 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*.

#### Task 22 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 Node Manager and Administration Server, and the URL is needed to access the Administration Server.

Click **Finish** to dismiss the configuration wizard.

## 10.7 Creating the System Components on BIHOST1

Perform the steps in this section to create the BI Cluster Controller, BI Scheduler, BI Presentation Services, and BI JavaHost system components on BIHOST1.

> **Note:**
>
> Replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.

1. Start WLST:

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

   ```
   ./wlst.sh
   ```

2. Open the BI Administration Server domain for updating:

   ```
   wls:/offline>readDomain('ASERVER_HOME')
   ```

3. Create a new BI Cluster Controller system component:

```
wls:/offline/
bi_domain>createOBICCSComponent('ASERVER_HOME','BIHOST1',port='OBICCS__port',por
tMonitor='OBICCS_monitor_port')
```

For example:

```
wls:/offline/bi_domain>createOBICCSComponent('/u01/oracle/config/domains/
bi_domain','BIHOST1',port='10006',portMonitor='10007')
```

4. Create a new BI Scheduler system component:

```
wls:/offline/
bi_domain>createOBISCHComponent('ASERVER_HOME','BIHOST1',port='OBISCH_port',port
Monitor='OBISCH_monitor_port')
```

For example:

```
wls:/offline/bi_domain>createOBISCHComponent('/u01/oracle/config/domains/
bi_domain','BIHOST1',port='10008',portMonitor='10009')
```

5. Create a new BI Presentation Services system component:

```
wls:/offline/
bi_domain>createOBIPSComponent('ASERVER_HOME','BIHOST1',port='OBIPS_port')
```

For example:

```
wls:/offline/bi_domain>createOBIPSComponent('/u01/oracle/config/domains/
bi_domain','BIHOST1',port='10010')
```

6. Create a new BI JavaHost system component:

```
wls:/offline/
bi_domain>createOBIJHComponent('ASERVER_HOME','BIHOST1',port='OBIJH_port')
```

For example:

```
wls:/offline/bi_domain>createOBIJHComponent('/u01/oracle/config/domains/
bi_domain','BIHOST1',port='10011')
```

7. Update and save the domain:

```
wls:/offline/bi_domain/SystemComponent/obijh1>updateDomain()
```

8. Close the domain:

```
wls:/offline/bi_domain/SystemComponent/obijh1>closeDomain()
```

9. Synchronize the changes with the WebLogic Server JDBC connection pools. This updates the midtier schema endpoints stored outside of WebLogic Server (for example, odbc.ini).

```
wls:/offline>syncMidtierDb('ASERVER_HOME')
```

10. Exit WLST:

```
wls:/offline>exit()
```

## 10.8 Creating a BI Service Instance

Perform the steps in this section to create a new BI Service instance.

**Note:**

Replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.

1. Start WLST:

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

./wlst.sh
```

2. Open the BI Administration Server domain for updating:

```
wls:/offline>readDomain('ASERVER_HOME')
```

3. Run the following command to create a new BI Service instance:

```
wls:/offline/bi_domain>createBIServiceInstance('ASERVER_HOME',
'service1',owner='weblogic',
description='Default Service', bar='ORACLE_HOME/bi/bifoundation/samples/
sampleapplite/SampleAppLite.bar',
port='OBIS_port', portMonitor='OBIS_monitor_port',machine='BIHOST1')
```

For example:

```
wls:/offline/bi_domain>createBIServiceInstance('/u01/oracle/config/domains/
bi_domain','service1',owner='weblogic',
description='Default Service', bar='/u01/oracle/products/fmw/bi/bifoundation/
samples/sampleapplite/SampleAppLite.bar',
port='10020', portMonitor='10021',machine='BIHOST1')
```

4. Update and save the domain:

```
wls:/offline/bi_domain/SystemComponent/obis1>updateDomain()
```

5. Close the domain for editing:

```
wls:/offline/bi_domain/SystemComponent/obis1>closeDomain()
```

6. Optional: Configure the default key to allow standard URLs to be used for the BI Service instance:

```
wls:/offline>f = open('ASERVER_HOME/config/fmwconfig/biconfig/bi-security/
config.properties', 'a')
wls:/offline>f.write ("defaultServiceInstanceKey=" + 'service1')
wls:/offline>f.close()
```

7. `SampleAppLite` uses XML data, which needs to be unzipped to the correct location.

```
import os
import zipfile
def unzip(srcZip, destDir):
    # no native unzip-to-dest in jython 2.2 :-(
    z = zipfile.ZipFile(srcZip)
```

```
    if not destDir.endswith('/'):
        destDir += '/'
    if not os.path.exists(destDir):
        os.makedirs(destDir)
    for f in z.namelist():
        outpath = destDir + f
        if f.endswith('/'):
            os.makedirs(outpath)
        else:
            outfile = open(outpath, 'wb')
            outfile.write(z.read(f))
            outfile.close()

wls:/offline>srcZip='ORACLE_HOME/bi/bifoundation/samples/sampleapplite/
SampleAppLite-datafiles.zip'
wls:/offline>destDir='ASERVER_HOME/bidata/service_instances/service1/data'
wls:/offline>unzip(srcZip,destDir)
```

8. Exit WLST:

```
exit()
```

## 10.9 Configuring the Singleton Data Directory (SDD)

Oracle Business Intelligence metadata is stored in a Singleton Data Directory (SDD). Metadata is managed in an Oracle Business Intelligence archive (BAR) file containing information about the Presentation Catalog, the metadata repository, and security authentication.

Perform the following steps to set up a shared directory for the Singleton Data Directory:

---

**Note:** The path to the Singleton Data Directory (SDD) is defined in the `ASERVER_HOME/config/fmwconfig/bienv/core/bi-environment.xml` file.

---

1. Create a shared directory for the Singleton Data Directory (SDD):

   For example:

   ```
   mkdir Shared_Storage_Location/biconfig
   ```

2. Move the data in the `ASERVER_HOME/bidata` directory to the shared directory you just created:

   ```
   mv ASERVER_HOME/bidata Shared_Storage_Location/biconfig
   ```

3. Update the Singleton Data Directory location in the `bi-environment.xml` file by doing the following:

   a. Open the `ASERVER_HOME/config/fmwconfig/bienv/core/bi-environment.xml` file for editing.

   b. Edit the file to change the Singleton Data Directory location from the default `$DOMAIN_HOME/bidata` directory to the absolute path of the shared `bidata` directory.

   For example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<bi:environment xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:bi="http://
bi.oracle.com/lcm">
     <bi:logoff-url></bi:logoff-url>
     <bi:singleton-data-directory>Shared_Storage_Location/biconfig/bidata</
bi:singleton-data-directory>
     <bi:services-domain-name-suffix>bi.outsourcing.com</bi:services-domain-name-
suffix>
</bi:environment>
```

4. Save and close the file.

## 10.10 Configuring Security for Essbase in Oracle Business Intelligence

Essbase components installed using the Oracle Business Intelligence installer cannot use Native Essbase or Hyperion Shared Services (HSS) security.

However, when you install Essbase with Oracle Business Intelligence, the Common Security Service (CSS) token-based identity assertion continues to be available and enables Oracle Business Intelligence to connect to Essbase data sources (both Essbase installed with Oracle Business Intelligence and Essbase installed with Enterprise Performance Management (EPM)) with the credentials of the end user. For this mechanism to work with an Essbase data source external to the Oracle Business Intelligence installation, you must follow the documentation. Also note that if multiple Essbase data sources are being used by Oracle Business Intelligence and there is a requirement to use this mechanism, all Essbase data sources must use the same shared secret for producing CSS tokens.

For more information, see Configuring Oracle Business Intelligence to Use Hyperion SSO Tokens when Communicating with Essbase, Hyperion Financial Management, Hyperion Planning in the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

To configure Common Security Services (CSS) security for Essbase components, you must run the following after creating the initial BI domain and before starting the domain:

```
cd ASERVER_HOME/bitools/bin
./generate_css_secrets.sh
```

## 10.11 Configuring the Domain Directories and Starting the Servers on BIHOST1

After the domain is created, you must perform a series of additional configuration tasks on BIHOST1. For example, you start the Node Manager and Administration Server. You then create a separate domain directory for the Managed Server. In this new and separate Managed Server directory, you start a second Node Manager instance and start the Managed Server and the Business Intelligence system components.

**See Also:**

Starting the Node Manager in the Administration Server Domain Home on BIHOST1
: Use these steps to start the per-domain Node Manager for the *ASERVER_HOME* domain directory.

Creating the boot.properties File
: You must create a `boot.properties` if you want start the Node Manager without being prompted for the Node Manager credentials. This step is required in an enterprise deployment. The credentials you enter in this file are encrypted when you start the Administration Server.

Starting the Administration Server
: Use these steps to start the Administration Server using the Node Manager.

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 Serverls.

Disabling the Derby Database

Creating a Separate Domain Directory for Managed Servers on BIHOST1
: 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 both BIHOST1 and BIHOST2. 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 BIHOST1

Starting the WLS_BI1 Managed Server on BIHOST1
: Use Oracle Enterprise Manager Fusion Middleware Control to start the Managed Server on BIHOST1.

Starting the System Components
: Use Oracle Enterprise Manager Fusion Middleware Control to start the system components for Oracle Business Intelligence.

## 10.11.1 Starting the Node Manager in the Administration Server Domain Home on BIHOST1

Use these steps to start the per-domain Node Manager for the *ASERVER_HOME* domain directory.

1. Verify that the listen address in the `nodemanager.properties` file is set correctly:

   a. Open the following file, using a text editor:

   ```
   ASERVER_HOME/nodemanager/nodemanager.properties
   ```

b. Make sure the `ListenAddress` property is set to the value of the ADMINVHN virtual IP address.

2. Navigate to the following directory:

```
ASERVER_HOME/bin
```

3. Use the following command to start the Node Manager:

```
nohup ./startNodeManager.sh > ASERVER_HOME/nodemanager/nodemanager.out 2>&1 &
```

For more information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

## 10.11.2 Creating the boot.properties File

You must create a `boot.properties` if you want start the Node Manager without being prompted for the Node Manager credentials. This step is required in an enterprise deployment. The credentials you enter in this file are encrypted when you start the Administration Server.

To create a `boot.properties` file for the Administration Server:

1. Create the following directory structure:

```
mkdir -p ASERVER_HOME/servers/AdminServer/security
```

2. In a text editor, create a file called `boot.properties` in the `security` directory 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 get 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 get encrypted.

3. Save the file and close the editor.

## 10.11.3 Starting the Administration Server

Use these steps to start the Administration Server using the Node Manager.

1. Start WLST:

```
cd ORACLE_COMMON_HOME/common/bin
./wlst.sh
```

2. Connect to Node Manager using the Node Manager credentials you defined in when you created the domain in the Configuration Wizard:

```
wls:/offline>nmConnect('nodemanager_username','nodemanager_password',
         'ADMINVHN','5556','domain_name',
         'ASERVER_HOME')
```

**Note:**

This username and password are used only to authenticate connections between Node Manager and clients. They are independent of the server admin ID and password and are stored in the `nm_password.properties` file located in the following directory:

*ASERVER_HOME*/config/nodemanager

3. Start the Administration Server:

   ```
   nmStart('AdminServer')
   ```

4. Exit WLST:

   ```
   exit()
   ```

## 10.11.4 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 Serverls.

To navigate to Fusion Middleware Control, enter the following URL, and log in with the Oracle WebLogic Server administrator credentials:

*ADMINVHN*:7001/em

To navigate to the Oracle WebLogic Server Administration Console, enter the following URL, and log in with the same administration credentials:

*ADMINVHN*:7001/console

## 10.11.5 Disabling the Derby Database

Before you create the Managed Server directory and start the Managed Servers, 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 will start automatically when you start the Managed Servers.

To disable the Derby database:

1. Navigate to the following directory in the Oracle home.

   *WL_HOME*/common/derby/lib

2. Rename the Derber library jar file:

   ```
   mv derby.jar disable_derby.jar
   ```

### 10.11.6 Creating a Separate Domain Directory for Managed Servers on BIHOST1

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 both BIHOST1 and BIHOST2. The domain directory on the local (or private) storage will be used to run the Managed Servers.

Placing the 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 tmp 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 ASERVER_HOME variable, and the path to the Managed Server domain home is represented by the MSERVER_HOME variable.

To create the Managed Server domain directory:

1. Log in to BIHOST1 and run the `pack` command to create a template as follows:

   ```
   cd ORACLE_COMMON_HOME/common/bin

   ./pack.sh -managed=true
             -domain=ASERVER_HOME
             -template=complete_path/bidomaintemplate.jar
             -template_name=bi_domain_template
   ```

   In this example:

   - Replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.
   - Replace *complete_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.
   - `bidomaintemplate.jar` is a sample name for the jar file you are creating, which will contain the domain configuration files.
   - `bi_domain_template` is the name assigned to the domain template file.

2. Make a note of the location of the `bidomaintemplate.jar` file you just created with the pack command.

   You must specify a full path for the template jar file as part of the `-template` argument to the pack command:

   ```
   ORACLE_COMMON_HOME/common/bin/
   ```

   > **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 BIHOST1 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=MSERVER_HOME \
            -overwrite_domain=true \
            -template=complete_path/bidomaintemplate.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 setUserOverrides.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 *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 *complete_path* with the complete path to the location where you created or copied the template jar file.
- `bidomaintemplate.jar` is the name of the template jar file you created when you ran the pack command to pack up the domain on the shared storage device.

**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 BIHOST1 local storage device.

### 10.11.7 Starting the Node Manager in the Managed Server Domain Directory on BIHOST1

After you create the Managed Server domain directory, there are two domain home directories and two corresponding Node Manager instances on BIHOST1. 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.

Follow these steps to start the Node Manager from the Managed Server home:

1. Navigate to the following directory:

   *MSERVER_HOME*/bin

2. Use the following command to start the Node Manager:

   ```
   nohup ./startNodeManager.sh > ./nodemanager.out 2>&1 &
   ```

For information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

### 10.11.8 Starting the WLS_BI1 Managed Server on BIHOST1

Use Oracle Enterprise Manager Fusion Middleware Control to start the Managed Server on BIHOST1.

Fusion Middleware Control is available because you already started the Node Manager and Administration Server in a previous step:

1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

   http://*ADMINVHN*:7001/em

   In this example:

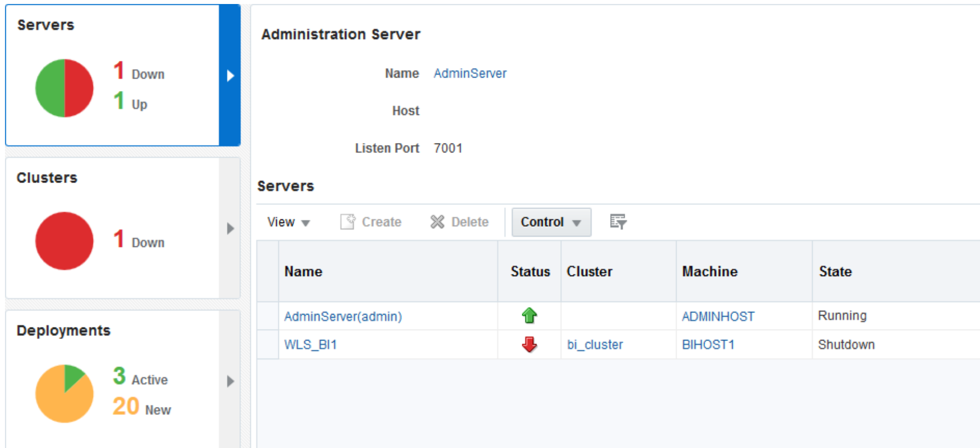
   - Replace *ADMINVHN* with the host name assigned to the ADMINVHN Virtual IP address.
   - Port 7001 is the typical port used for the Administration Server console and Fusion Middleware Control. However, you should use the actual URL that was displayed at the end of the Configuration Wizard session when you created the domain.

     **Tip:**

     For more information about managing Oracle Fusion Middleware using Oracle Enterprise Manager Fusion Middleware Control, see "Getting Started Using Oracle Enterprise Manager Fusion Middleware Control" in *Administering Oracle Fusion Middleware*.

2. Log in to Fusion Middleware Control using the Administration Server credentials.

3. Select the **Servers** pane to view the Managed Servers in the domain.




4. Select only the **WLS_BI1** Managed Server and click **Control** on the tool bar. Then, under **Control**, select **Start**.

## 10.11.9 Starting the System Components

Use Oracle Enterprise Manager Fusion Middleware Control to start the system components for Oracle Business Intelligence.

Fusion Middleware Control is already available because you already started the Node Manager and the Administration Server in a previous step.

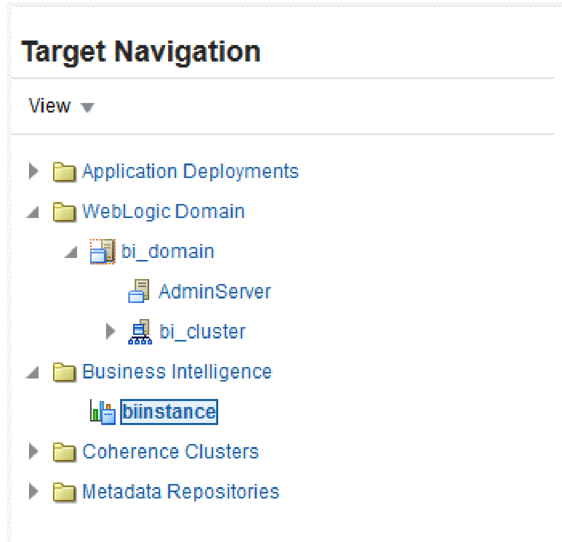
1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

   ```
   http://ADMINVHN:7001/em
   ```

2. Log into Fusion Middleware Control using the Administration Server credentials.

3. If not already displayed, click the Target Navigation icon  in the top left corner of the page to display the **Target Navigation** pane.

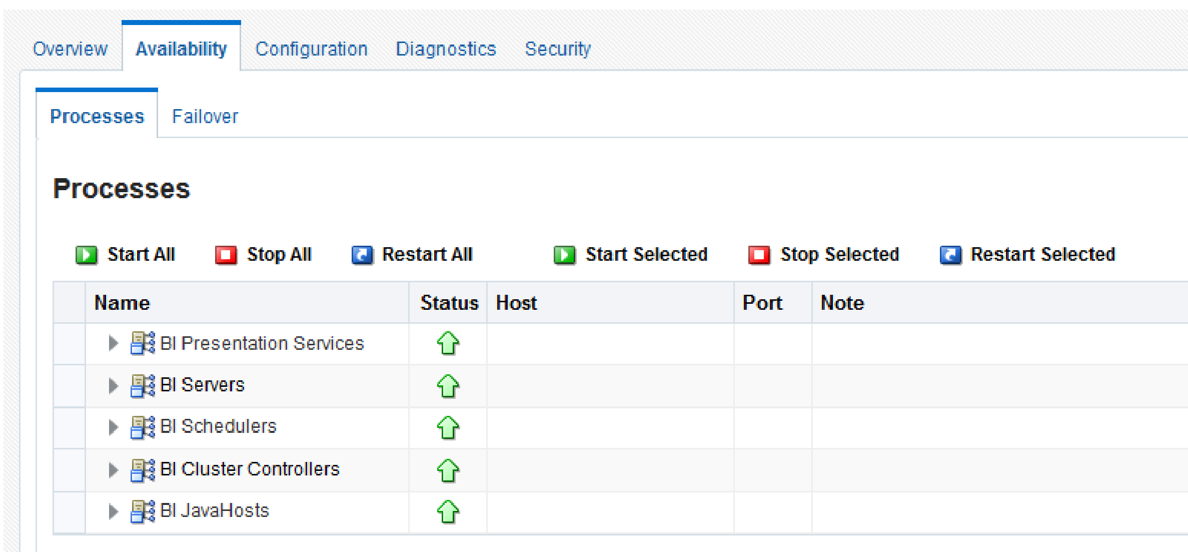
4. In the **Target Navigation** pane, expand the **Business Intelligence** folder and select **biinstance**.




The Business Intelligence Overview page appears.

5. Click **Availability** and then **Processes** to display the **Processes** tab on the Availability page.

6. Click **Start All** to start all the components.




## 10.12 Setting Up the Global Cache

The global cache is a query cache that is shared by all Oracle BI servers participating in a cluster. It is recommended that you configure the global cache so that cache seeding and purging events can be shared by all Oracle BI servers participating in a cluster.

For more information about the global cache, see About the Global Cache in the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

To set up the global cache:

1. Create a shared directory for the global cache.

```
mkdir Shared_Storage_Location/global_cache
```

All Oracle BI servers must have read and write access to this directory.

2. Use the Performance tab of the Configuration page in Fusion Middleware Control to set the **Global cache path** and **Global cache size**.

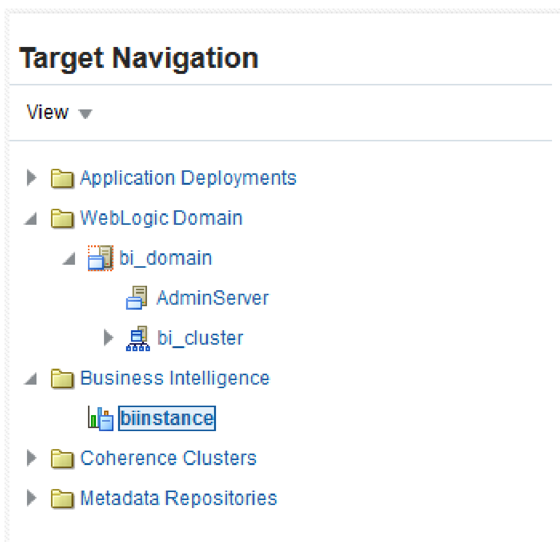
   a. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

      `http://ADMINVHN:7001/em`

   b. Log in to Fusion Middleware Control using the Administration Server credentials.

   c. If not already displayed, click the Target Navigation icon  in the top left corner of the page to display the **Target Navigation** pane.

   d. In the **Target Navigation** pane, expand the **Business Intelligence** folder and select **biinstance**.




   The Business Intelligence Overview page appears.

   e. Click **Configuration** and then **Performance** to display the Performance tab of the Configuration page.

   f. Click **Lock & Edit** in the Change Center menu at the top right corner of the page.

   g. Specify the shared directory you created for storing purging and seeding cache entries in the **Global cache path** field. Enter a value for the **Global cache size** to specify the maximum size of the global cache (for example, 250 MB).

   h. Click **Apply**.

   i. Click **Activate Changes** in the Change Center menu at the top right corner of the page.

## 10.13 Verifying Oracle Business Intelligence URLs on BIHOST1

After starting the components in the domain on BIHOST1, access these URLs to verify the configuration of Oracle Business Intelligence.

- Access the following URL to verify the status of WLS_BI1:

  ```
  http://BIHOST1VHN1:7003/analytics
  ```

  You will be redirected to:

  ```
  http://bi.example.com/analytics
  ```

- Access the following URL to verify the status of the BI Publisher application:

  ```
  http://BIHOST1VHN1:7003/xmlpserver
  ```

  You will be redirected to:

  ```
  http://bi.example.com/xmlpserver
  ```

- Access the following URL to verify the status of the Oracle Essbase application:

  ```
  http://BIHOST1VHN1:7003/aps/Essbase
  ```

## 10.14 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 you have already installed and configured a supported LDAP directory, such as Oracle Unified Directory or Oracle Internet Directory.

**See Also:**

About the Supported Authentication Providers

About the Enterprise Deployment Users and Groups

Prerequisites for Creating a New Authentication Provider and Provisioning Users and Groups

Provisioning a Domain Connector User in the LDAP Directory

Creating the New Authentication Provider

Provisioning an Enterprise Deployment Administration User and Group

Adding the New Administration User to the Administration Group

Updating the boot.properties File and Restarting the System

## 10.14.1 About the Supported Authentication Providers

Oracle Fusion Middleware supports a variety of LDAP authentication providers. For more information, see "Identity Store Types and WebLogic Authenticators" in *Securing Applications with Oracle Platform Security Services*.

The instructions in this guide assume you will be using one of the following providers:

- Oracle Unified Directory
- Oracle Internet Directory
- Oracle Virtual Directory

---

**Note:**

By default, the instructions here describe how to configure the identity service instance to support querying against a single LDAP identity store.

However, you can configure the service to support a virtualized identity store, which queries multiple LDAP identity stores, using LibOVD.

For more information about configuring a Multi-LDAP lookup, refer to "Configuring the Identity Store Service" in *Securing Applications with Oracle Platform Security Services*.

---

## 10.14.2 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.

**See Also:**

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 Business Intelligence

Example Users and Roles Used in This Guide

### 10.14.2.1 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.

Such an approach is not recommended. Instead, it is a best practice to assign a unique distinguished name (DN) within the directory tree for the users and groups you provision for the administration of your Oracle Fusion Middleware domains.

For example, if you plan to install and configure an Oracle Business Intelligence enterprise deployment domain, then create a user called **`weblogic_bi`** and an administration group called **`BI Administrators`**.

#### 10.14.2.2 About the Domain Connector User

Oracle recommends that you create a separate domain connector user (for example, `biLDAP`) 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 Manager do not reconcile this user.

#### 10.14.2.3 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.

To add new users to the central LDAP directory, you cannot use the WebLogic Administration Console. Instead, you must use the appropriate LDAP modification tools, such as ldapbrowser or JXplorer.

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.

Enabling virtualization involves the following steps:

1. Locate and open the following configuration file with a text editor:

   ```
   DOMAIN_HOME/config/fmwconfig/jps-config.xml
   ```

2. Add or update the following property, as follows:

   ```
   <property name="virtualize" value="true"/>
   ```

   For more information about the virtualize property, see “OPSS System and Configuration Properties” in *Securing Applications with Oracle Platform Security Services*.

#### 10.14.2.4 About Product-Specific Roles and Groups for Oracle Business Intelligence

Each Oracle Fusion Middleware product implements its own predefined roles and groups for administration and monitoring.

As a result, as you extend the domain to add additional products, you can add these product-specific roles to the `BI Administrators` group. After they are added to the `BI Administrators` group, each product administrator user can administer the domain with the same set of privileges for performing administration tasks.

Instructions for adding additional roles to the `BI Administrators` group are provided in Common Configuration and Management Tasks for an Enterprise Deployment.

#### 10.14.2.5 Example Users and Roles Used in This Guide

In this guide, the examples assume that you provision the following administration user and group with the DNs shown below:

- Admin User DN:

```
cn=weblogic_bi,cn=users,dc=example,dc=com
```

- Admin Group DN:

```
cn=BI Administrators,cn=groups,dc=example,dc=com
```

- Product-specific LDAP Connector User:

```
cn=biLDAP,cn=systemids,dc=example,dc=com
```

This is the user you will 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
```

---

**Note:**

When using Oracle Unified Directory, this user will need 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
```

---

## 10.14.3 Prerequisites for Creating a New Authentication Provider and Provisioning Users and Groups

Before you create a new LDAP authentication provider, 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
```

In addition, back up the `boot.properties` file for the Administration Server in the following directory:

```
DOMAIN_HOME/servers/AdminServer/security
```

## 10.14.4 Provisioning a Domain Connector User in the LDAP Directory

This example shows how to create a user called `biLDAP` in the central LDAP directory.

To provision the user in the LDAP provider:

1. Create an ldif file named `domain_user.ldif` with the contents shown below and then save the file:

```
dn: cn=biLDAP,cn=systemids,dc=example,dc=com
changetype: add
orclsamaccountname: biLDAP
userpassword: password
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
```

```
objectclass: orcluser
objectclass: orcluserV2
mail: biLDAP@example.com
givenname: biLDAP
sn: biLDAP
cn: biLDAP
uid: biLDAP
```

---

**Note:**

If you are using Oracle Unified Directory, then add the following four group memberships to the end of the LDIF file to grant the appropriate read/write privileges:

```
dn:
cn=orclFAUserReadPrivilegeGroup,cn=groups,dc=us,dc=oracle,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=biLDAP,cn=systemids,dc=us,dc=oracle,dc=com

dn: cn=orclFAGroupReadPrivilegeGroup,cn=groups,dc=us,dc=oracle,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=biLDAP,cn=systemids,dc=us,dc=oracle,dc=com

dn: cn=orclFAUserWritePrivilegeGroup,cn=groups,dc=example,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=biLDAP,cn=systemids,dc=example,dc=com

dn: cn=orclFAGroupWritePrivilegeGroup,cn=groups,dc=example,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=biLDAP,cn=systemids,dc=example,dc=com
```

---

2. Provision the user in the LDAP directory.

   For example, for an Oracle Unified Directory LDAP provider:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a \
                                 -h oudhost.example.com
                                 -D "cn=oudadmin" \
                                 -w password \
                                 -p 1389 \
                                 -f domain_user.ldif
```

   For Oracle Internet Directory:

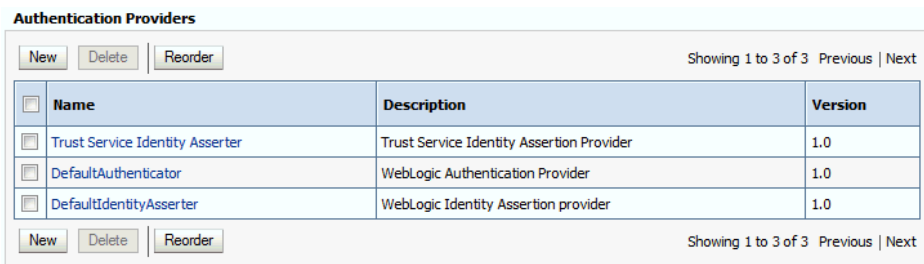
```
OID_ORACLE_HOME/bin/ldapadd -h oidhost.example.com \
                            -p 3060 \
                            -D cn="orcladmin" \
                            -w password \
                            -c \
                            -v \
                            -f domain_user.ldif
```

### 10.14.5 Creating the New Authentication Provider

To configure a new LDAP-based authentication provider:

1. Log in to the WebLogic Server Administration Console.
2. Click **Security Realms** in the left navigational bar.
3. Click the **myrealm** default realm entry.
4. Click the **Providers** tab.

   Note that there is a `DefaultAuthenticator` provider configured for the realm. This is the default WebLogic Server authentication provider.

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
   - `OIDAuthenticator` for Oracle Internet Directory
   - `OVDAuthenticator` for Oracle Virtual Directory

8. Select the authenticator type from the **Type** drop-down list.

   Select one of the following types, based on the LDAP directory service you are planning to use as your credential store:

   - `OracleUnifiedDirectoryAuthenticator` for Oracle Unified Directory
   - `OracleInternetDirectoryAuthenticator` for Oracle Internet Directory
   - `OracleVirtualDirectoryAuthenticator` for Oracle Virtual 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 save the control flag settings.

13. Click the **Provider Specific** tab and enter the details specific to your LDAP server, as shown in the following table.

    Note that 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*.

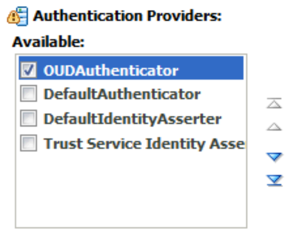
| Parameter | Sample Value | Value Description |
|---|---|---|
| Host | For example: `oud.example.com` | The LDAP server's server ID. |
| Port | For example: `1689` | The LDAP server's port number. |
| Principal | For example: `cn=biLDAP, cn=systemids,dc=example,dc=com` | The LDAP user DN used to connect 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. |

| Parameter | Sample Value | Value Description |
|---|---|---|
| All Users Filter | `(&(uid=*)(objectclass=person))` | Instead of a default search criteria for **All Users Filter**, search all users based on the `uid` value.<br>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.<br>For example, if the **User Name Attribute** type is `cn`, then this field should be set to:<br>`(&(cn=*)(objectclass=person)))` |
| User From Name Filter | For example:<br>`(&(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.<br>For example, if the **User Name Attribute** type is `cn`, then this field should be set to:<br>`(&(cn=%u)(objectclass=person)))`. |
| User Name Attribute | For example: `uid` | The attribute of an LDAP user object that specifies the name of the user. |
| Group Base DN | For example: `cn=groups,dc=example,dc=com` | Specify the DN that points to your Groups node. |
| Use Retrieved User Name as Principal | Checked | Must be turned on. |
| 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 then use the resulting page to make the Provider you just created first in the list of authentication providers.




17. Click **OK**.

18. In the Change Center, click **Activate Changes** .

19. Restart the Administration Server and all managed servers.

    To stop the Managed Servers, log in to Fusion Middleware Control, select the Managed Servers in the Target Navigator and click **Shut Down** in the toolbar.

    To stop and start the Administration Server using the Node Manager:

    a. Start WLST:

    ```
    cd ORACLE_COMMON_HOME/common/bin
    ./wlst.sh
    ```

    b. Connect to Node Manager using the Node Manager credentials you defined in when you created the domain in the Configuration Wizard:

    ```
    wls:/offline>nmConnect('nodemanager_username','nodemanager_password',
                'ADMINVHN','5556','domain_name',
                'ASERVER_HOME')
    ```

    c. Stop the Administration Server:

    ```
    nmKill('AdminServer')
    ```

    d. Start the Administration Server:

    ```
    nmStart('AdminServer')
    ```

    e. Exit WLST:

    ```
    exit()
    ```

    To start the Managed Servers, log in to Fusion Middleware Control, select the Managed Servers, and click **Start Up** in the toolbar.

20. After the restart, review the contents of the following log file:

    ```
    ASERVER_HOME/servers/AdminServer/logs/AdminServer.log
    ```

    Verify that no LDAP connection errors occurred. For example, look for errors such as the following:

    ```
    The LDAP authentication provider named "OUDAuthenticator" failed to make
    connection to ldap server at ...
    ```

    If you see such errors in the log file, then check the authorization provider connection details to verify they are correct and try saving and restarting the Administration Server again.

21. After you restart and verify that no LDAP connection errors are in the log file, try browsing the users and groups that exist in the LDAP provider:

In the Administration Console, navigate to the **Security Realms > myrealm > Users and Groups** page. You should be able to see all users and groups that exist in the LDAP provider structure.

### 10.14.6 Provisioning an Enterprise Deployment Administration User and Group

This example shows how to create a user called **`weblogic_bi`** and a group called **`BI Administrators`**.

To provision the administration user and group in LDAP provider:

1. Create an ldif file named `admin_user.ldif` with the contents shown below and then save the file:

```
dn: cn=weblogic_bi,cn=users,dc=example,dc=com
changetype: add
orclsamaccountname: weblogic_bi
userpassword: password
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
objectclass: orcluser
objectclass: orcluserV2
mail: weblogic_bi@example.com
givenname: weblogic_bi
sn: weblogic_bi
cn: weblogic_bi
uid: weblogic_bi
```

2. Provision the user in the LDAP directory.

   For example, for an Oracle Unified Directory LDAP provider:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a \
                                 -h oudhost.example.com
                                 -D "cn=oudadmin" \
                                 -w password \
                                 -p 1389 \
                                 -f admin_user.ldif
```

   For Oracle Internet Directory:

```
OID_ORACLE_HOME/bin/ldapadd -h oidhost.example.com \
                            -p 3060 \
                            -D cn="orcladmin" \
                            -w password \
                            -c \
                            -v \
                            -f admin_user.ldif
```

3. Create an `ldif` file named `admin_group.ldif` with the contents shown below and then save the file:

```
dn: cn=WCCAdministrators,cn=Groups,dc=example,dc=com
displayname: WCCAdministrators
objectclass: top
objectclass: WCCAdministrators
objectclass: orclGroup
uniquemember: cn=weblogic_wcc,cn=users,dc=example,dc=com
```

```
cn: WCCAdministrators
description: Administrators Group for the Oracle WebCenter Content Domain
```

4. Provision the group in the LDAP Directory.

   For Oracle Unified Directory:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a \
                                 -D "cn=oudadmin" \
                                 -h oudhost.example.com \
                                 -w password \
                                 -p 3060 \
                                 -f admin_group.ldif
```

   For Oracle Internet Directory:

```
OID_ORACLE_HOME/bin/ldapadd -h oid.example.com \
                            -p 3060 \
                            -D cn="orcladmin" \
                            -w password \
                            -c \
                            -v \
                            -f admin_group.ldif
```

5. Verify that the changes were made successfully:

   a. Log in to the Oracle WebLogic Server Administration Console.

   b. In the left pane of the console, click **Security Realms**.

   c. Click the default security realm (**myrealm**).

   d. Click the **Users and Groups** tab.

   e. Verify that the administrator user and group you provisioned are listed on the page.

### 10.14.7 Adding the New Administration User to the Administration Group

After adding the users and groups to Oracle Internet Directory, the group must be assigned 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 using the administration credentials that you provided in the Configuration Wizard.

   Do not use the credentials for the administration user 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** button.

8. Select **Group** from the **Predicate List** drop-down menu, and then click **Next**.

9. Enter `BI Administrators` in the **Group Argument Name** field, and then click **Add**.

   `BI Administrators` is added to the list box of arguments.

10. Click **Finish** to return to the Edit Global Role page.

    The `BI Administrators` group is now listed.

11. Click **Save** to finish adding the **Admin** Role to the `BI Administrators` group.

12. Validate that the changes were made by logging in to the WebLogic Administration Server Console using the new `weblogic_bi` 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 you just provisioned in the new authentication provider, then you have configured the provider successfully.

### 10.14.8 Updating the boot.properties File and Restarting the System

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 BIHOST1, go the following directory:

   ```
   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_bi
   password=password
   ```

5. Save the file.

6. Restart the Administration Server.

## 10.15 Backing Up the Oracle Business Intelligence Configuration

It is an Oracle best practices recommendation to create a backup after successfully configuring a domain or at another logical point. Create a backup after verifying 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.

# 11

# Configuring the Web Tier for an Enterprise Deployment

This chapter describes how to install and configure a standalone Oracle HTTP Server domain that contains two Oracle HTTP Server instances: one on WEBHOST1 and one on WEBHOST2.

This chapter provides information on variables used when configuring the web tier and installing and configuring a web tier domain.

**See Also:**

Configuring the Enterprise Deployment

allows for a simple configuration that requires a minimum amount of configuration and a minimum amount of resouces to run and maintain.

Installing Oracle HTTP Server on WEBHOST1
The following sections describe how to install the Oracle HTTP Server software on the web tier.

Creating a Web Tier Domain on WEBHOST1
The following sections describe how to create a new Oracle HTTP Server standalone domain on the first Web tier host.

Installing and Configuring a Web Tier Domain on WEBHOST2
After you install Oracle HTTP Server and configure a Web Tier 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
The following sections describe how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.

Configuring Oracle HTTP Server to Route Requests to the Application Tier
The following sections describe how to update the Oracle HTTP Server configuration files so the web server instances route requests to the servers in the domain.

Backing Up the Configuration
It is an Oracle best practices recommendation to create a backup after successfully configuring a domain or at another logical point. Create a backup after verifying 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.

## 11.1 Variables Used When Configuring the Web Tier

As you perform the tasks in this chapter, you will be referencing 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.

- *OHS_ORACLE_HOME*
- *OHS_DOMAIN_HOME*

In addition, you'll be referencing the following virtual IP (VIP) address and host names:

- ADMINVHN
- WEBHOST1
- WEBHOST2

## 11.2 About the Web Tier 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 resouces to run and maintain.

For more information about the role and configuration of the Oracle HTTP Server instances in the web tier, see Understanding the Web Tier.

# 11.3 Installing Oracle HTTP Server on WEBHOST1

The following sections describe how to install the Oracle HTTP Server software on the web tier.

**See Also:**

Starting the Installer on WEBHOST1

Navigating the Oracle HTTP Server Installation Screens

Verifying the Oracle HTTP Server Installation

## 11.3.1 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. Launch the installation program by entering the following command:

   ```
   ./fmw_12.2.1.0.0_ohs_linux64.bin
   ```

   When the installation program appears, you are ready to begin the installation.

## 11.3.2 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 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.<br>For the purposes of an enterprise deployment, enter the value of the OHS_ORACLE_HOME variable listed in Table 7-3. |

| Screen | Description |
|---|---|
| Installation Type | Select **Standalone HTTP Server (Managed independently of WebLogic server)**.<br>This installation type allows you to configure the Oracle HTTP Server instances independently from any other existing Oracle WebLogic Server domains. |
| Prerequisite Checks | This screen verifies that your system meets the minimum necessary requirements.<br>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. |
| Security Updates | If you already have an Oracle Support account, use this screen to indicate how you would like to receive security updates.<br>If you do not have an account, or if you are sure you want to skip this step, then clear the check box and verify your selection in the follow-up dialog box. |
| Installation Summary | Use this screen to verify the installation options 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.<br>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. |

## 11.3.3 Verifying the Oracle HTTP Server Installation

To verify that your Oracle HTTP Server installation completed successfully, list files that were installed in the new Oracle home directory. You should see the following directories in the Oracle HTTP Server Oracle home:

```
bin
cfgtoollogs
crs
css
has
install
inventory
```

```
jlib
ldap
lib
network
nls
ohs
OPatch
oracle_common
oracore
oraInst.loc
oui
perl
plsql
plugins
precomp
rdbms
root.sh
slax
sqlplus
srvm
webgate
wlserver
xdk
```

## 11.4 Creating a Web Tier Domain on WEBHOST1

The following sections describe how to create a new Oracle HTTP Server standalone domain on the first Web tier host.

**See Also:**

Starting the Configuration Wizard on WEBHOST1

Navigating the Configuration Wizard Screens for a Web Tier Domain

### 11.4.1 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 OHS_ORACLE_HOME/oracle_common/common/bin
./config.sh
```

### 11.4.2 Navigating the Configuration Wizard Screens for a Web Tier 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:

- 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, "Adding System Components"
- Task 5, "OHS Server Screen"

- Task 6, "Reviewing Your Configuration Specifications and Configuring the Domain"
- Task 7, "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 OHS_DOMAIN_HOME variable.

Note the following:

- The Configuration Wizard will create the new directory that you specify here.
- Create the directory on local storage, so the web servers do not have any dependencies on on storage devices outside the DMZ.

---

**[other]:**

- More information about the Domain home directory can be found in "Choosing a Domain Home" in *Planning an Installation of Oracle Fusion Middleware*.
- More information about the other options on this screen can be found in "Configuration Type" in *Creating WebLogic Domains Using the Configuration Wizard*.
- For more information about the Web tier and the DMZ, see Understanding the Firewalls and Zones of a Typical Enterprise Deployment.
- For more information about the OHS_DOMAIN_HOME directory variable, see File System and Directory Variables Used in This Guide.

---

**Task 2 Selecting the Configuration Templates**

On the Templates screen, select **Oracle HTTP Server (Standalone) - 12.2.1.0 [ohs]**.

**Tip:**
More information about the options on this screen can be found in "Templates" in *Creating WebLogic Domains Using the Configuration Wizard*.

**Task 3 Selecting the JDK for the Web Tier Domain.**

Select the Oracle Hotspot JDK, which was installed in the Web tier Oracle home when you installed the Oracle HTTP Server software.

**Task 4 Adding System Components**

On the System Components screen, create one Oracle HTTP Server instance. The screen should by default have a single instance defined.

1. Note that the default instance name is `ohs1` in the **System Component** field. You use this default name.

2. Make sure `OHS` is selected in the **Component Type** field.

3. Use the **Restart Interval Seconds** field to specify the number of seconds to wait before attempting 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 OHS Server Screen

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 of the remaining fields are pre-populated, but you can change the values as required for your organization. For more information about the fields on this screen, see "OHS Server" in *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:

   ```
   WEBHOST1:7777
   ```

#### Task 6 Configuring Node Manager

Select **Per Domain Default Location** as the Node Manager type, and specify the Node Manager credentials.

---

**Note:**

More information about the options on this screen can be found in "Node Manager" in *Creating Domains Using the Configuration Wizard*.

More information about the Node Manager types can be found in "Node Manager Overview" in *Administering Node Manager for Oracle WebLogic Server*.

---

#### Task 7 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the detailed configuration information for the domain you are about to extend. 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*.

#### Task 8 Writing Down Your Domain Home

The Configuration Success screen will show the updated domain home location.

Make a note of the information provided here, as you will need it to start the servers and access the Administration Server.

Click **Finish** to dismiss the configuration wizard.

## 11.5 Installing and Configuring a Web Tier Domain on WEBHOST2

After you install Oracle HTTP Server and configure a Web Tier domain on WEBHOST1, then you must also perform the same tasks on WEBHOST2.

1. Log in to WEBHOST2 and install Oracle HTTP Server, using the instructions in Installing Oracle HTTP Server on WEBHOST1.

2. Configure a new standalone domain on WEBHOST2, 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.

## 11.6 Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2

The following sections describe how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.

**See Also:**

Starting the Node Manager on WEBHOST1 and WEBHOST2

Starting the Oracle HTTP Server Instances

### 11.6.1 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:

   ```
   OHS_DOMAIN_HOME/bin
   ```

2. Start the Node Manager as shown below, using `nohup` and `nodemanager.out` as an example output file:

   ```
   nohup OHS_DOMAIN_HOME/bin/startNodeManager.sh > OHS_DOMAIN_HOME/nodemanager/
   nodemanager.out 2>&1 &
   ```

3. Log in to WEBHOST2 and perform steps 1 and 2.

For more information about additional Node Manager configuration options, see *Administering Node Manager for Oracle WebLogic Server*.

### 11.6.2 Starting the Oracle HTTP Server Instances

To start the Oracle HTTP Server instances:

1. Navigate to the following directory on WEBHOST1:

   ```
   OHS_DOMAIN_HOME/bin
   ```

For more information about the location of the OHS_DOMAIN_HOME directory, see File System and Directory Variables Used in This Guide.

2. Enter the following command:

```
./startComponent.sh ohs1
```

3. When prompted, enter the Node Manager password.

4. Repeat steps 1 through 3 to start the `ohs2` instance on WEBHOST2.

For more information, see "Starting Oracle HTTP Server Instances" in *Administering Oracle HTTP Server*.

## 11.7 Configuring Oracle HTTP Server to Route Requests to the Application Tier

The following sections describe how to update the Oracle HTTP Server configuration files so the web server instances route requests to the servers in the domain.

**See Also:**

About the Oracle HTTP Server Configuration for an Enterprise Deployment

Modifying the httpd.conf File to Include Virtual Host Configuration Files

Creating the Virtual Host Configuration Files for Oracle Business Intelligence

Validating the Virtual Server Configuration on the Load Balancer

Validating Access to the Management Consoles and Administration Server

Validating Access to the HTTP Access to the Business Intelligence Components

### 11.7.1 About the Oracle HTTP Server Configuration for an Enterprise Deployment

The following topics provide overview information about the changes required to the Oracle HTTP Server configuration in an enterprise deployment.

**See Also:**

Purpose of the Oracle HTTP Server Virtual Hosts

Recommended Structure of the Oracle HTTP Server Configuration Files

#### 11.7.1.1 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.

#### 11.7.1.2 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 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).

### 11.7.2 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 OHS_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
   ```

3. Open the `httpd.conf` file in a text editor and make the following changes:

   a. Create a `NameVirtualHost` entry as follows immediately below the comment: "# Use name-based virtual hosting" and before the "# VirtualHost example" comment:

      ```
      NameVirtualHost WEBHOST1:7777
      ```

      In this example, replace WEBHOST1 with the value of the WEBHOST1 variable. For more information, see File System and Directory Variables Used in This Guide.

   b. Verify that there is an `INCLUDE` statement in the `httpd.conf` that includes all `*.conf` files in the `moduleconf` subdirectory:

      ```
      IncludeOptional "moduleconf/*.conf"
      ```

      This statement makes it possible to create the separate virtual host files for each component, making it easier to update, maintain, and scale out the virtual host definitions.

4. Save the `httpd.conf` file.
5. Log in to WEBHOST2 and perform steps 2 through 4 to update the `httpd.conf` file for the `ohs2` instance.

   On WEBHOST2, replace all instances of WEBHOST1 with WEBHOST2 and all instances of `ohs1` with `ohs2`.

### 11.7.3 Creating the Virtual Host Configuration Files for Oracle Business Intelligence

Use the following procedure to create the required configuration files so the Oracle HTTP Server routes requests to the Oracle Business Intelligence servers.

---

**Note:** Before you create the virtual host configuration files, be sure you have configured the virtual servers on the load balancer, as described in Purpose of the Oracle HTTP Server Virtual Hosts.

---

To create the virtual host configuration files:

1. Log in to WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (ohs1):

   ```
   cd OHS_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
   ```

2. Create the `admin_vh.conf` file and add the following directive:

   ```
   <VirtualHost *:7777>
      ServerName admin.example.com:80
      ServerAdmin you@your.address
      RewriteEngine On
      RewriteOptions inherit
      </VirtualHost>

   # Admin Server and EM
   <Location /console>
       SetHandler weblogic-handler
       WebLogicHost ADMINVHN
       WeblogicPort 7001
   </Location>

   <Location /consolehelp>
       SetHandler weblogic-handler
       WebLogicHost ADMINVHN
       WeblogicPort 7001
   </Location>

   <Location /em>
       SetHandler weblogic-handler
       WebLogicHost ADMINVHN
       WeblogicPort 7001
   </Location>
   ```

3. Create the `biinternal_vh.conf` file and add the following directives:

   ```
   <VirtualHost WEBHOST1:7777>
       ServerName biinternal.example.com
       ServerAdmin you@your.address
       RewriteEngine On
       RewriteOptions inherit
   </VirtualHost>

   #redirect browser requests that omit document/dir
   RedirectMatch 301 /analytics$ /analytics/
   RedirectMatch 301 /biservices$ /biservices/
   RedirectMatch 301 /analytics-ws$ /analytics-ws/
   RedirectMatch 301 /AdminService$ /AdminService/
   RedirectMatch 301 /AsyncAdminService$ /AsyncAdminService/
   RedirectMatch 301 /wsm-pm$ /wsm-pm/
   RedirectMatch 301 /xmlpserver$ /xmlpserver/
   RedirectMatch 301 /bisearch$ /bisearch/
   RedirectMatch 301 /mapviewer$ /mapviewer/
   RedirectMatch 301 /va$ /va/
   RedirectMatch 301 /bicomposer$ /bicomposer/
   ```

```
RedirectMatch 301 /mobile$ /mobile/
RedirectMatch 301 /aps$ /aps/
RedirectMatch 301 /bi-security$ /bi-security/
RedirectMatch 301 /workspace$ /workspace/

# WSM-PM
<Location /wsm-pm>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# BIEE Analytics
<Location /analytics>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

<Location /bicontent>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

<Location /mobile>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

<Location /va>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# MapViewer
<Location /mapviewer>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# BI Publisher
<Location /xmlpserver>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# BI Search
<Location /bisearch>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# BI Composer
<Location /bicomposer>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# EPM Provider Services
<Location /aps>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>
```

```
# EPM Workspace
<Location /workspace>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# BI SOA Services
<Location /biservices>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# AdminService
<Location /AdminService>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

# AsyncAdminService
<Location /AsyncAdminService>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

#OWSM
<Location /wsm-pm>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
</Location>

#BI Security
<Location /bi-security>
    SetHandler weblogic-handler
    WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
    </Location>
```

4. Create the `bi_vh.conf` file and add the following directives

```
<VirtualHost *:7777>
   ServerName https://bi.example.com:443
   ServerAdmin you@your.address
   RewriteEngine On
   RewriteOptions inherit
</VirtualHost>

#redirect browser requests that omit document/dir
RedirectMatch 301 /analytics$ /analytics/
RedirectMatch 301 /xmlpserver$ /xmlpserver/
RedirectMatch 301 /analytics/res$ /analytics/res/
RedirectMatch 301 /biofficeclient$ /biofficeclient/
RedirectMatch 301 /biservices$ /biservices/
RedirectMatch 301 /analytics-ws$ /analytics-ws/
RedirectMatch 301 /wsm-pm$ /wsm-pm/
RedirectMatch 301 /bisearch$ /bisearch/
RedirectMatch 301 /mapviewer$ /mapviewer/
RedirectMatch 301 /va$ /va/
RedirectMatch 301 /bicontent$ /bicontent/
RedirectMatch 301 /bicomposer$ /bicomposer/
RedirectMatch 301 /mobile$ /mobile/
RedirectMatch 301 /aps$ /aps/
```

```
RedirectMatch 301 /workspace$ /workspace/

# BIEE Analytics
<Location /analytics>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /analytics-ws>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /bicontent>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /mobile>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /va>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location

# MapViewer
<Location /mapviewer>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# BI Publisher
<Location /xmlpserver>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# BI Search
<Location /bisearch>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

```
# BI Composer

<Location /bicomposer>
   SetHandler weblogic-handler
   WebLogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# EPM Provider Services
<Location /aps>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# EPM Workspace
<Location /workspace>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# OWSM
<Location /wsm-pm>
   SetHandler weblogic-handler
   WeblogicCluster APPHOST1VHN1:7003,APPHOST2VHN1:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

5. Restart the ohs1 instance:

   a. Change directory to the following location:

      ```
      cd OHS_DOMAIN_HOME/bin
      ```

   b. Enter the following commands to stop and start the instance; provide the node manager password when prompted:

      ```
      ./stopComponent.sh ohs1
      ./startComponent.sh ohs1
      ```

6. Copy the three .conf files (`admin_vh.conf`, `biinternal_vh.conf`, and `biinternal_vh.conf`) to the configuration directory for the second Oracle HTTP Server instance (ohs2) on WEBHOST2:

   ```
   OHS_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/module_conf
   ```

7. Edit the .conf files and change any references from WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.

8. Restart the ohs2 instance:

   a. Change directory to the following location:

      ```
      cd OHS_DOMAIN_HOME/bin
      ```

   b. Enter the following commands to stop and start the instance:

```
./stopComponent.sh ohs2
./startComponnet.sh ohs2
```

### 11.7.4 Validating the Virtual Server Configuration on the Load Balancer

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.

- `http://admin.example.com/index.html`
- `http://biinternal.example.com/index.html`
- `http://bi.example.com/index.html`

### 11.7.5 Validating Access to the Management Consoles and Administration Server

To verify the changes you have made in this chapter:

1. Use the following URL to the hardware load balancer to display the Oracle WebLogic Server Administration Console, and log in using the Oracle WebLogic Server administrator credentials:

   ```
   http://admin.example.com/console
   ```

   This validates that the `admin.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.

2. Similarly, you should be able to access the Fusion Middleware Control using a similar URL:

   ```
   http://admin.example.com/em
   ```

### 11.7.6 Validating Access to the HTTP Access to the Business Intelligence Components

After you configure the Oracle HTTP Server instances, you can validate your work by accessing key Oracle Business Intelligence URLs. If these URLs display the proper content, then you can be assured the Web tier components are configured correctly.

To validate HTTP access to the Oracle Business Intelligence components, enter each of the following URLs in your Web browser and make sure the proper content displays:

- http://bi.example.com/analytics
- http://bi.example.com/mapviewer
- http://bi.example.com/xmlpserver
- http://biinternal.example.com/wsm-pm
- http://bi.example.com/bicomposer
- http://bi.example.com/aps/Essbase
- http://bi.example.com/aps/SmartView

## 11.8 Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after successfully configuring a domain or at another logical point. Create a backup after verifying 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.

# 12

# Scaling Out Oracle Business Intelligence

This chapter describes the steps to scale out your initial Oracle Business Intelligence domain to BIHOST2.

Scaling out your components involves installing the Oracle Business Intelligence on the other host computers, stopping and cloning the components on BIHOST1, packing and unpacking the domain and starting the components after scaling out.

**See Also:**

Configuring the Enterprise Deployment

Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers

Installing Oracle Business Intelligence on the Other Host Computers
: If you have configured a separate shared storage volume or partition for BIHOST2, then you must also install the software on BIHOST2.

Stopping the Components on BIHOST1
: Before you scale out, you must stop all the component processes in the domain on BIHOST1.

Cloning the Components on BIHOST1
: After stopping all the component processes, you must clone the components in the domain on BIHOST1, which will create new components on BIHOST2 based on the initial domain you created.

Packing Up the Initial Domain on BIHOST1
: Use the steps in this section to create a template jar file that contains the domain configuration information, which now includes configuration information about the Oracle HTTP Server instances.

Unpacking the Domain on BIHOST2
: Use the steps in this section to unpack the domain template containing the domain configuration information and copy the `bidomaintemplate.jar` file from BIHOST1 to BIHOST2.

Starting the Components on BIHOST1 and BIHOST2 After Scaling Out
: This section describes the steps to start the component processes on BIHOST1 and BIHOST2 after scale out. The component processes

include the Node Managers, the Administration Server for the WebLogic domain, the system components, and the Managed Servers.

Verifying Oracle Business Intelligence URLs on BIHOST2
After starting the components in the domain on BIHOST2, access these URLs to verify the configuration of Oracle Business Intelligence.

Configuring Oracle BI Publisher
Perform these manual tasks to configure Oracle BI Publisher.

Backing Up the Oracle Business Intelligence Configuration After Scaling Out
It is an Oracle best practices recommendation to create a backup after successfully configuring a domain or at another logical point. Create a backup after verifying 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.

## 12.1 Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers

If you have configured a separate shared storage volume or partition for BIHOST2, then you must also install the Infrastructure on BIHOST2.

For more information, 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 on BIHOST1 and Navigating the Infrastructure Installation Screens to create the Oracle home on the appropriate storage device.

---

**Note:**

In previous releases, the recommended enterprise topology included a colocated set of Oracle HTTP Server instances. In those releases, there was a requirement to install the Infrastructure on the Web Tier hosts (WEBHOST1 and WEBHOST2). However, for this release, the enterprise deployment topology assumes the Web servers are installed and configured in standalone mode, so they are not considered part of the application tier domain. For more information, see Configuring the Web Tier for an Enterprise Deployment

---

## 12.2 Installing Oracle Business Intelligence on the Other Host Computers

If you have configured a separate shared storage volume or partition for BIHOST2, then you must also install the software on BIHOST2.

For more information, 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 Installation Program and Navigating the Installation Screens.

# 12.3 Stopping the Components on BIHOST1

Before you scale out, you must stop all the component processes in the domain on BIHOST1.

The component processes include the Node Managers, the Administration Server for the WebLogic domain, the system components, and the WLS_BI1 Managed Server, which is controlled by the Node Manager.

To stop all the components in the domain on BIHOST1, perform the following tasks.

**See Also:**


Stopping the System Components

Stopping the WLS_BI1 Managed Server

Stopping the Administration Server

Stopping the Node Manager in the Administration Server Domain Home

Stopping the Node Manager in the Managed Server Domain Directory


## 12.3.1 Stopping the System Components

Follow these steps to stop the system components using Fusion Middleware Control.

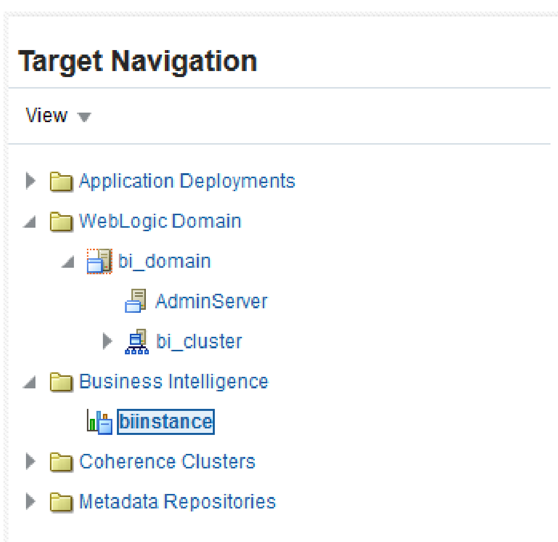
1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

```
http://ADMINVHN:7001/em
```

2. Log into Fusion Middleware Control using the Administration Server credentials.

3. If not already displayed, click the Target Navigation icon in the top left corner of the page to display the **Target Navigation** pane.

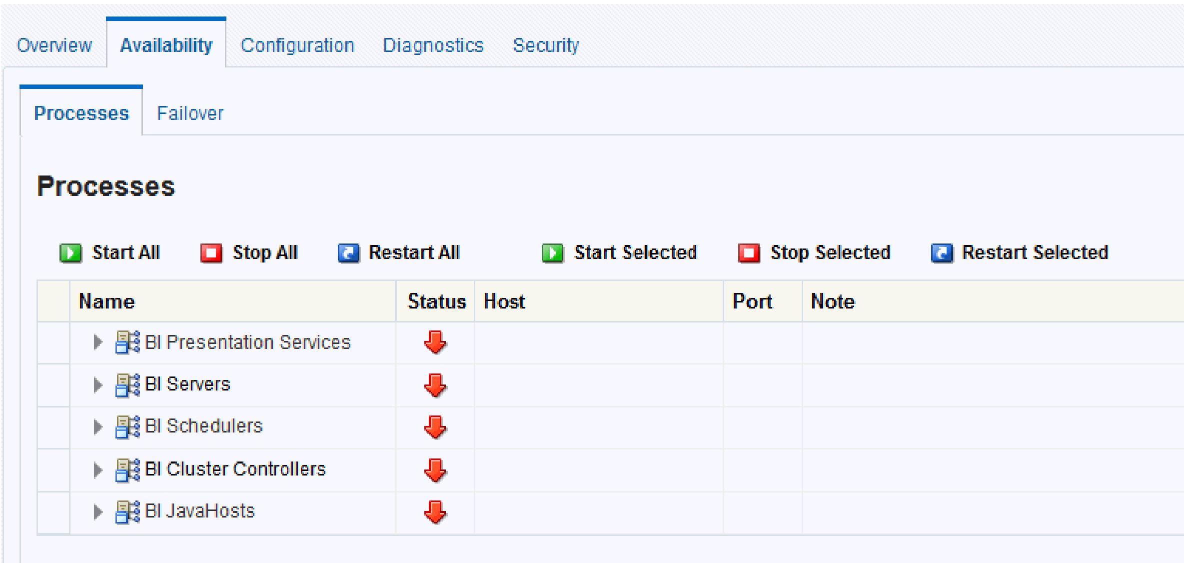
4. In the **Target Navigation** pane, expand the **Business Intelligence** folder and select **biinstance**.



The Business Intelligence Overview page appears.

5. Click **Availability** and then **Processes** to display the **Processes** tab on the Availability page.

6. Click **Stop All** to stop all the system components.




## 12.3.2 Stopping the WLS_BI1 Managed Server

Follow these steps to stop the WLS_BI1 Managed Server using Fusion Middleware Control.

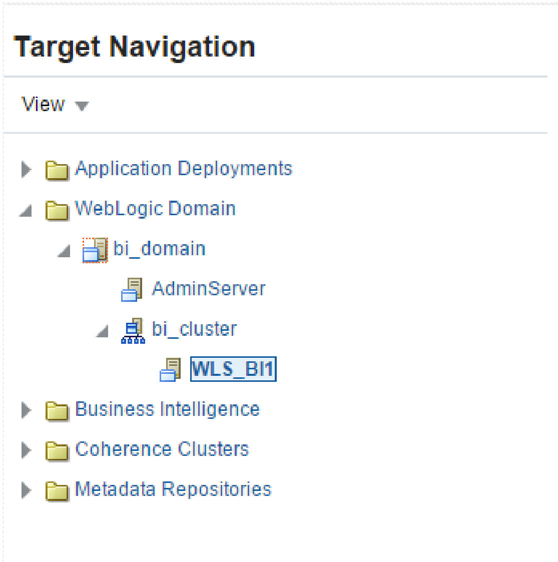
1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:

   `http://ADMINVHN:7001/em`

2. Log in to Fusion Middleware Control using the Administration Server credentials.

3. If not already displayed, click the Target Navigation icon in the top left corner of the page to display the **Target Navigation** pane.

4. In the **Target Navigation** pane, expand the domain under the **WebLogic Domain** folder to view the Managed Servers in the domain.




5. Select only the **WLS_BI1** Managed Server.

6. Click **Shut Down...** on the Oracle WebLogic Server toolbar.

7. When the shut down operation is complete, navigate to the Domain home page and verify that the WLS_BI1 Managed Server is down.

## 12.3.3 Stopping the Administration Server

Use these steps to stop the Administration Server using the Node Manager.

1. Start WLST:

```
cd ORACLE_COMMON_HOME/common/bin
./wlst.sh
```

2. Connect to Node Manager using the Node Manager credentials you defined in when you created the domain in the Configuration Wizard:

```
wls:/offline>nmConnect('nodemanager_username','nodemanager_password',
            'ADMINVHN','5556','domain_name',
            'ASERVER_HOME')
```

---

**Note:**

This username and password are used only to authenticate connections between Node Manager and clients. They are independent of the server admin ID and password and are stored in the `nm_password.properties` file located in the following directory:

```
ASERVER_HOME/config/nodemanager
```

---

3. Stop the Administration Server:

```
nmKill('AdminServer')
```

4. Exit WLST:

```
exit()
```

### 12.3.4 Stopping the Node Manager in the Administration Server Domain Home

Use these steps to stop the per-domain Node Manager for the *ASERVER_HOME* domain directory.

1. Navigate to the following directory:

   ```
   ASERVER_HOME/bin
   ```

2. Use the following command to stop the Node Manager:

   ```
   ./stopNodeManager.sh
   ```

### 12.3.5 Stopping the Node Manager in the Managed Server Domain Directory

Use these steps to stop the per-domain Node Manager for the *MSERVER_HOME* domain directory.

1. Navigate to the following directory:

   ```
   MSERVER_HOME/bin
   ```

2. Use the following command to stop the Node Manager:

   ```
   ./stopNodeManager.sh
   ```

## 12.4 Cloning the Components on BIHOST1

After stopping all the component processes, you must clone the components in the domain on BIHOST1, which will create new components on BIHOST2 based on the initial domain you created.

Perform the following steps on BIHOST1 to create additional components by cloning your existing Managed Server, Node Manager, system components, and service instance. You will later pack and unpack the new components on BIHOST2.

1. Start WLST:

   ```
   cd ORACLE_HOME/oracle_common/common/bin
   ./wlst.sh
   ```

2. Open the BI Administration Server domain for updating:

   ```
   wls:/offline> readDomain('ASERVER_HOME')
   ```

   In this example, replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.

3. Run the `CloneBIMachine` command to create additional components based on the existing components in the initial BI domain:

   ```
   wls:/offline/bi_domain>
   cloneBIMachine('ASERVER_HOME','bihost2.example.com',baseMachine='BIHOST1',baseServ
   er='WLS_BI1',machineName='BIHOST2')
   ```

   In this example:

   - Replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.

- Replace `bihost2.example.com` with the listen address of the new machine, BIHOST2.
- `WLS_BI1` is the server name used throughout this document for the BI Managed Server on BIHOST1. If you chose a different name, be sure to replace it as needed.

4. Update and save the domain:

```
wls:/offline/bi_domain/SystemComponent/obis2> updateDomain()
```

5. Close the domain:

```
wls:/offline/bi_domain/SystemComponent/obis2> closeDomain()
```

6. Exit WLST:

```
wls:/offline> exit()
```

## 12.5 Packing Up the Initial Domain on BIHOST1

Use the steps in this section to create a template jar file that contains the domain configuration information, which now includes configuration information about the Oracle HTTP Server instances.

1. Log in to BIHOST1 and run the `pack` command to create a template jar file as follows:

```
cd ORACLE_COMMON_HOME/common/bin

./pack.sh -managed=true \
          -domain=ASERVER_HOME \
          -template=complete_path/bidomaintemplate.jar \
          -template_name=bi_domain_template
```

   In this example:

   - Replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.
   - Replace *complete_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.
   - `bidomaintemplate.jar` is a sample name for the jar file you are creating, which will contain the domain configuration files, including the configuration files for the Oracle HTTP Server instances.
   - `bi_domain_template` is the name assigned to the domain template file.

2. Make a note of the location of the `bidomaintemplate.jar` file you just created with the pack command.

   By default, the pack template file is created in the current directory where you ran the pack command. In this example, it would be created in the *ORACLE_COMMON_HOME*`/common/bin` directory, but you can specify a full path for the template jar file as part of the `-template` argument to 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*.

## 12.6 Unpacking the Domain on BIHOST2

Use the steps in this section to unpack the domain template containing the domain configuration information and copy the `bidomaintemplate.jar` file from BIHOST1 to BIHOST2.

1. Log in to BIHOST2.
2. Copy the domain template jar file from BIHOST1 to BIHOST2.
3. If you haven't already, create the recommended directory structure for the Managed Server domain on the BIHOST2 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:

   *complete_path*

   ```
   cd ORACLE_COMMON_HOME/common/bin
   ```

   ```
   ./unpack.sh -domain=MSERVER_HOME \
       -template=complete_path/bidomaintemplate.jar \
       -app_dir=APPLICATION_HOME \
       -overwrite_domain=true \
       -nodemanager_type=PerDomainNodeManager
   ```

   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 will be unpacked.
   - Replace *complete_path* with the complete path to the location where you created or copied the template jar file.
   - `bidomaintemplate.jar` is the directory path and name of the template you created when you ran the pack command to pack up the domain.

     Note that if you are using a separate shared storage volume or partition for BIHOST2 (and redundant Oracle homes), then you must first copy the template to the volume or partition mounted to BIHOST2.
   - Replace ***`APPLICATION_HOME`*** with the complete path to the Application directory for the domain on shared storage.

   **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 `MSERVER_HOME` directory and verify that the domain configuration files were copied to the correct location on the BIHOST2 local storage device.

## 12.7 Starting the Components on BIHOST1 and BIHOST2 After Scaling Out

This section describes the steps to start the component processes on BIHOST1 and BIHOST2 after scale out. The component processes include the Node Managers, the Administration Server for the WebLogic domain, the system components, and the Managed Servers.

To start the components after scale out, perform the following tasks.

**See Also:**

Starting the Node Manager in the Administration Server Domain Home

Starting the Administration Server

Starting the Node Managers in the Managed Server Domain Directories

Setting the Listen Address for the WLS_BI2 Managed Server

Configuring the WebLogic Proxy Plug-In

Starting the Managed Servers

Starting the System Components

### 12.7.1 Starting the Node Manager in the Administration Server Domain Home

To start the per-domain Node Manager for the *ASERVER_HOME* domain directory on BIHOST1, use the procedure in Starting the Node Manager in the Administration Server Domain Home on BIHOST1.

### 12.7.2 Starting the Administration Server

To start the Administration Server using the Node Manager, use the procedure in Starting the Administration Server.

### 12.7.3 Starting the Node Managers in the Managed Server Domain Directories

To start the per-domain Node Managers for the *MSERVER_HOME* directories on BIHOST1 and BIHOST2, use the procedure in Starting the Node Manager in the Managed Server Domain Directory on BIHOST1.

### 12.7.4 Setting the Listen Address for the WLS_BI2 Managed Server

Before you can start the WLS_BI2 Managed Server, you must update the listen address of the Managed Server so that it is set to the value of the BIHOST2VHN virtual IP address. To do this, you use the WebLogic Server Administration Console.

1. Log in to the WebLogic Server Administration Console using your WebLogic administrator credentials.

2. Click **Lock & Edit** in the Change Center.

3. Expand the **Environment** node in the **Domain Structure** menu on the left and then click **Servers**.
4. In the Servers table, click the name of the Managed Server (**WLS_BI2**).
5. In the Configuration tab, set the **Listen Address** to the value of BIHOST2VHN and then click **Save** to save your changes.
6. Click **Activate Changes** in the Change Center.

### 12.7.5 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 for the clusters you just configured.

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Domain Structure** pane, expand the **Environment** node.
3. Click **Clusters**.
4. Select the cluster to which you want to proxy requests from Oracle HTTP Server.

   The **Configuration: General** tab is displayed.
5. Scroll down to the **Advanced** section and expand it.
6. Click **Lock & Edit** in the Change Center.
7. Set **WebLogic Plug-In Enabled** to **yes**.
8. Click **Save** and click **Activate Changes**.
9. Click **Activate Changes** in the Change Center.

### 12.7.6 Starting the Managed Servers

Before starting the WLS_BI2 Managed Server, make sure that the listen address of WLS_BI2 is set to the value of the BIHOST2VHN virtual IP address. For more information, see Setting the Listen Address for the WLS_BI2 Managed Server.

To start the WLS_BI1 and WLS_BI2 Managed Servers, use the procedure in Starting the WLS_BI1 Managed Server on BIHOST1.

### 12.7.7 Starting the System Components

To start all of the Oracle Business Intelligence system components, use the procedure in Starting the System Components.

## 12.8 Verifying Oracle Business Intelligence URLs on BIHOST2

After starting the components in the domain on BIHOST2, access these URLs to verify the configuration of Oracle Business Intelligence.

- Access the following URL to verify the status of WLS_BI2:

```
http://BIHOST2VHN1:7003/analytics
```

You will be redirected to:

```
http://bi.example.com/analytics
```

- Access the following URL to verify the status of the BI Publisher application:

  ```
  http://BIHOST2VHN1:7003/xmlpserver
  ```

  You will be redirected to:

  ```
  http://bi.example.com/xmlpserver
  ```

- Access the following URL to verify the status of the Oracle Essbase application:

  ```
  http://BIHOST2VHN1:7003/aps/Essbase
  ```

## 12.9 Configuring Oracle BI Publisher

Perform these manual tasks to configure Oracle BI Publisher.

**See Also:**

Updating the JMS Shared Temp Directory
: Follow these steps to update the JMS Shared Temp Directory for BI Publisher Scheduler. You need to perform the steps in this section on only one of the BI hosts (either BIHOST1 or BIHOST2).

Setting the Oracle BI EE Data Source
: The Oracle Business Intelligence Enterprise Edition (BI EE) data source must point to the clustered Oracle BI Servers through the Cluster Controllers. You must perform this task in BI Publisher.

### 12.9.1 Updating the JMS Shared Temp Directory

Follow these steps to update the JMS Shared Temp Directory for BI Publisher Scheduler. You need to perform the steps in this section on only one of the BI hosts (either BIHOST1 or BIHOST2).

Perform the following steps to update the BI Publisher Scheduler configuration:

1. Log in to BI Publisher using one of the following URLs:

   - http://BIHOST1VHN1:7003/xmlpserver
   - http://BIHOST2VHN1:7003/xmlpserver

2. Click the **Administration** tab.

3. Click **Scheduler Configuration** under System Maintenance.

   The Scheduler Configuration screen is displayed.

4. Update the **Shared Directory** by entering a directory that is located in the shared storage. This shared storage is accessible from both BIHOST1 and BIHOST2.

5. Click **Test JMS**.

   You see a confirmation message that JMS tested successfully.

---

**Note:**

If you do not see a confirmation message for a successful test, then verify that the JNDI URL is set to the following:

```
cluster:t3://bi_cluster
```

---

6. Click **Apply**.
7. Check the Scheduler status from the **Scheduler Diagnostics** tab.

### 12.9.2 Setting the Oracle BI EE Data Source

The Oracle Business Intelligence Enterprise Edition (BI EE) data source must point to the clustered Oracle BI Servers through the Cluster Controllers. You must perform this task in BI Publisher.

Perform the following steps to set the Oracle BI EE data source in BI Publisher:

1. Log in to BI Publisher using the following URL with administrator credentials.

   ```
   http://BIHOST1VHN1:7003/xmlpserver
   ```

2. Select the **Administration** tab.
3. Under **Data Sources**, select **JDBC Connection**.
4. Update the Oracle BI EE data source setting by changing the **Connection String** parameter to the following:

   ```
   jdbc:oraclebi://primary_cluster_controller_host:primary_cluster_controller_
   port/PrimaryCCS=primary_cluster_controller_host;PrimaryCCSPort=primary_cluster_
   controller_port;SecondaryCCS=secondary_cluster_controller_host
   ;SecondaryCCSPort=secondary_cluster_controller_port
   ```

   For example:

   ```
   jdbc:oraclebi://BIHOST1:9706/PrimaryCCS=BIHOST1;PrimaryCCSPort=9706;
   SecondaryCCS=BIHOST2;SecondaryCCSPort=9706;
   ```

5. Select **Use System User**.

   If you do not want to use the system user for the connection, then deselect **Use System User** and specify the BIImpersonateUser credentials for **Username** and **Password**. For more information about the BIImpersonateUser user in this context, see Credentials for Connecting to the Oracle BI Presentation Catalog in the *Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Enterprise Edition*.

6. Click **Test Connection**. You see a "Connection established successfully" message.
7. Click **Apply**.

## 12.10 Backing Up the Oracle Business Intelligence Configuration After Scaling Out

It is an Oracle best practices recommendation to create a backup after successfully configuring a domain or at another logical point. Create a backup after verifying 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.

# Part IV

## Common Configuration and Management Procedures for an Enterprise Deployment

The following topics contain configuration and management procedures that are required or recommended for a typical enterprise deployment.

**See Also:**

# 13

# Common Configuration and Management Tasks for an Enterprise Deployment

The following topics describe configuration and management tasks you will likely need to perform on the enterprise deployment environment.

**See Also:**

Common Configuration and Management Procedures for an Enterprise Deployment

Verifying Manual Failover of the Administration Server
: In case a host computer fails, you can fail over the Administration Server to another host. The following sections provide the steps to verify the failover and failback of the Administration Server from BIHOST1 and BIHOST2.

Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer
: This section describes how to enable SSL communication between the middle tier and the hardware load balancer.

Performing Backups and Recoveries for an Enterprise Deployment
: This section provides guidelines for making sure you back up the necessary directories and configuration data for an Oracle Business Intelligence enterprise deployment.

## 13.1 Verifying Manual Failover of the Administration Server

In case a host computer fails, you can fail over the Administration Server to another host. The following sections provide the steps to verify the failover and failback of the Administration Server from BIHOST1 and BIHOST2.

Assumptions:

- The Administration Server is configured to listen on ADMINVHN, and not on localhost or ANY 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 BIHOST1 to BIHOST2, and the two nodes have these IPs:

- BIHOST1: 100.200.140.165
- BIHOST2: 100.200.140.205
- ADMINVHN : 100.200.140.206. This is the Virtual IP where the Administration Server is running, assigned to ethX:Y, available in BIHOST1 and BIHOST2.

- Oracle WebLogic Server and Oracle Fusion Middleware components have been installed in BIHOST2 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 failover procedure.

**See Also:**

Failing Over the Administration Server to a Different Host
: The following procedure shows how to fail over the Administration Server to a different node (BIHOST2). Note that even after failover, the Administration Server will still use the same Oracle WebLogic Server *machine* (which is a logical machine, not a physical machine).

Validating Access to the Administration Server on BIHOST2 Through Oracle HTTP Server
: After you perform a manual failover of the Administation Server, it is important to verify that you can access the Administration Server, using the standard administration URLs.

Failing the Administration Server Back to BIHOST1
: After you have tested a manual Administration Server failover, and after you have validated that you can access the administration URLs after the failover, you can then migrate the Administration Server back to its original host.

### 13.1.1 Failing Over the Administration Server to a Different Host

The following procedure shows how to fail over the Administration Server to a different node (BIHOST2). Note that even after failover, the Administration Server will still use the same Oracle WebLogic Server *machine* (which is a logical machine, not a physical machine).

This procedure assumes you've configured a per domain Node Manager for the enterprise topology. For more information, see About the Node Manager Configuration in a Typical Enterprise Deployment

To fail over the Administration Server to a different host:

1. Stop the Administration Server.
2. Stop the Node Manager in the Administration Server domain directory (ASERVER_HOME).
3. Migrate the ADMINVHN virtual IP address to the second host:
   a. Run the following command as root on BIHOST1(where X:Y is the current interface used by ADMINVHN):

      ```
      /sbin/ifconfig ethX:Y down
      ```

b. Run the following command as root on BIHOST2:

```
/sbin/ifconfig <interface:index> ADMINVHN netmask <netmask>
```

For example:

```
/sbin/ifconfig eth0:1 100.200.140.206 netmask 255.255.255.0
```

---

**Note:**

Ensure that the netmask and interface to be used to match the available network configuration in BIHOST2.

---

4. Update the routing tables using `arping`, for example:

```
/sbin/arping -q -U -c 3 -I eth0 100.200.140.206
```

5. Start the Node Manager in the Administration Server domain home on BIHOST2.
6. Start the Administration Server on BIHOST2.
7. Test that you can access the Administration Server on BIHOST2 as follows:

   a. Ensure that you can access the Oracle WebLogic Server Administration Console using the following URL:

   ```
   http://ADMINVHN:7001/console
   ```

   b. Check that you can access and verify the status of components in Fusion Middleware Control using the following URL:

   ```
   http://ADMINVHN:7001/em
   ```

### 13.1.2 Validating Access to the Administration Server on BIHOST2 Through Oracle HTTP Server

After you perform a manual failover of the Administation Server, it is important to verify that you can access the Administration Server, 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 BIHOST2:

- 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.

### 13.1.3 Failing the Administration Server Back to BIHOST1

After you have tested a manual Administration Server failover, and after you have validated that you can access the administration URLs after the failover, you can then migrate the Administration Server back to its original host.

1. Stop the Administration Server.

2. Stop the Node Manager in the Administration Server domain home onBIHOST2.

3. Run the following command as root on BIHOST2.

   ```
   /sbin/ifconfig ethZ:N down
   ```

4. Run the following command as root on BIHOST1:

   ```
   /sbin/ifconfig ethX:Y 100.200.140.206 netmask 255.255.255.0
   ```

   **Note:**

   Ensure that the netmask and interface to be used match the available network configuration in BIHOST1

5. Update the routing tables using `arping` on BIHOST1:

   ```
   /sbin/arping -q -U -c 3 -I ethX 100.200.140.206
   ```

6. Start the Node Manager in the Administration Server domain home on BIHOST1.

7. Start the Administration Server on BIHOST1.

8. Test that you can access the Oracle WebLogic Server Administration Console using the following URL:

   ```
   http://ADMINVHN:7001/console
   ```

9. Check that you can access and verify the status of components in the Oracle Enterprise Manager using the following URL:

   ```
   http://ADMINVHN:7001/em
   ```

## 13.2 Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer

This section describes how to enable SSL communication between the middle tier and the hardware load balancer.

**Note:**

This steps are applicable if the hardware load balancer is configured with SSL and the front end address of the system has been secured accordingly.

**See Also:**

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

### 13.2.1 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 using the appropriate SSL configuration.

### 13.2.2 Generating Self-Signed Certificates Using the utils.CertGen Utility

This section describes the procedure for creating self-signed certificates on BIHOST1. Create these certificates using the network name or alias of the 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 using central or shared stores for the certificates used for different purposes (for example, SSL set up for HTTP invocations). In this case, BIHOST2 uses the cert directory created for BIHOST1certificates.

For information on using trust CA certificates instead, see the information about configuring identity and trust in *Oracle Fusion Middleware Securing 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. Set up your environment by running the `WL_HOME/server/bin/setWLSEnv.sh` script:

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 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
```

5. Change directory to the keystore home:

```
cd KEYSTORE_HOME
```

6. Run the `utils.CertGen` tool to create the certificates for both the physical host names and the virtual host names used by servers in the node.

   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 BIHOST1.example.com_cert \
      BIHOST1.example.com_key domestic BIHOST1.example.com
```

### 13.2.3 Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

This section describes how to create an Identity Keystore on BIHOST1.example.com.

In previous sections you have created certificates and keys that reside in a shared storage. In this section, the certificate and private key for both BIHOST1 and ADMINVHN are imported into a new Identity Store. Make sure that you use a different alias for each of the certificate/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 using the `utils.ImportPrivateKey` utility.

---

1. Import the certificate and private key for ADMINVHN and BIHOST1 into the Identity Store. Make sure that you use a different alias for each of the certificate/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:**

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/BIHOST1.example.com_cert.pem
     -keyfile KEYSTORE_HOME/BIHOST1.example.com_key.pem
     -keyfilepass password
     -keystore appIdentityKeyStore.jks
     -storepass password
     -alias BIHOST1
     -keypass password
```

2. Repeat the above step for all the remaining hosts used in the system (for example, BIHOST2).

### 13.2.4 Creating a Trust Keystore Using the Keytool Utility

To create the Trust Keystore on BIHOST1.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 always changing 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` 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
```

## 13.2.5 Importing the Load Balancer Certificate into the Trust Store

For the SSL handshake to behave properly, the load balancer's certificate needs to be added to the WLS servers trust store. For adding it, follow these steps:

1. Access the site on SSL with a browser (this add the server's certificate to the browser's repository).

2. From the browser's certificate management tool, export the certificate to a file that is on the server's file system (with a file name like `bi.example.com`).

3. Use the keytool to import the load balancer's certificate into the truststore:

```
keytool -import -file bi.example.com -v -keystore appTrustKeyStore.jks
```

## 13.2.6 Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts

To ensure each server can access the updated trust store, edit the setDomainEnv.sh script in each of the domain home directories in the enterprise deployment.

1. Log in to BIHOST1 and open the following file with a text editor:

```
ASERVER_HOME/bin/setDomainEnv.sh
```

2. Replace reference to the existing `DemoTrustStore` entry with the location of the `appTrustKeyStore.jks` file.

   Note that all the values for EXTRA_JAVA_PROPERTIES must be on one line in the file, followed by the export command on a new line. For example:

```
EXTRA_JAVA_PROPERTIES="${EXTRA_JAVA_PROPERTIES}
         -Djavax.net.ssl.trustStore=KEYSTORE_HOME/appTrustKeyStore.jks..."
export EXTRA_JAVA_PROPERTIES
```

3. Make the same change to the setDomainEnv.sh file in the *MSERVER_HOME*`/bin` directory on BIHOST1 and BIHOST2.

Alternatively, you can use the **`setUserOverrides.sh`** file to place these options, this way they will not get overwritten when the domain's scripts are extended with the configuration wizard or updated with unpack operations. For more information, see "Customizing Domain Wide Server Parameters" in *Administering Server Startup and Shutdown for Oracle WebLogic Server*.

### 13.2.7 Configuring Node Manager to Use the Custom Keystores

To configure the Node Manager to use the custom keystores, add the following lines to the end of the `nodemanager.properties` files located both in *`ASERVER_HOME/`*`nodemanager` and *`MSERVER_HOME/`*`nodemanager` directories in all nodes:

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=Identity KeyStore
CustomIdentityKeyStorePassPhrase=Identity KeyStore Passwd
CustomIdentityAlias=Identity Key Store Alias
CustomIdentityPrivateKeyPassPhrase=Private Key used when creating Certificate
```

Make sure to use the correct value for `CustomIdentityAlias` for Node Manager's listen address. For example on BIHOST1, use appIdentity2 according to the steps in Creating a Trust Keystore Using the Keytool Utility

```
(appIdentity2 mapped to the BIHOST1 listen address).
Example for Node 1:
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=KEYSTORE_HOME/appIdentityKeyStore.jks
CustomIdentityKeyStorePassPhrase=password
CustomIdentityAlias=appIdentity1
CustomIdentityPrivateKeyPassPhrase=password
```

The passphrase entries in the `nodemanager.properties` file are encrypted when you start Node Manager as described in "Starting the Node Manager on BIHOST1." For security reasons, minimize the time the entries in the `nodemanager.properties` file are left unencrypted. After you edit the file, start Node Manager as soon as possible so that the entries are encrypted.

### 13.2.8 Configuring WebLogic Servers to Use the Custom Keystores

Configure the WebLogic Servers to use the custom keystores 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. In the left pane, expand **Environment**, and select **Servers**.
3. Click the name of the server for which you want to configure the identity and trust keystores.
4. Select **Configuration**, and then **Keystores**.
5. In the **Keystores** field, click **Change**, and select **Custom Identity and Custom Trust** method for storing and managing private keys/digital certificate pairs and trusted CA certificates, and click Save.
6. 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.

7. 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 in as New_Password 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.

8. Click **Save**.

9. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

10. Click **Lock & Edit**.

11. Select **Configuration**, then **SSL**.

12. In the Private Key Alias field, enter the alias you used for the host name the managed server listens on.

    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

13. Click **Save**.

14. Click **Activate Changes** in the Administration Console's Change Center to make the changes take effect.

15. Restart the server for which the changes have been applied. The fact that servers can be restarted using the Administration Console/Node Manager is a good verification that the communication between Node Manager, Administration Server and the managed servers is correct.

## 13.3 Performing Backups and Recoveries for an Enterprise Deployment

This section provides guidelines for making sure you back up the necessary directories and configuration data for an Oracle Business Intelligence enterprise deployment.

**Note:**

Some of the static and run-time 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 13-1 lists the static artifacts to back up in a typical Oracle Business Intelligence enterprise deployment.

***Table 13-1 Static Artifacts to Back Up in the Oracle Business Intelligence 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 | BIHOST1 and BIHOST2 | Application Tier |
| Installation-related files | WEBHOST1, WEHOST2, and shared storage | N/A |

Table 13-2 lists the runtime artifacts to back up in a typical Oracle Business Intelligence enterprise deployment.

***Table 13-2 Run-Time Artifacts to Back Up in the Oracle Business Intelligence Enterprise Deployment***

| Type | Host | Tier |
|---|---|---|
| Administration Server domain home (ASERVER_HOME) | BIHOST1 and BIHOST2 | Application Tier |
| Application home (APPLICATION_HOME) | BIHOST1 and BIHOST2 | Application Tier |
| Oracle RAC databases | DBHOST1 and DBHOST2 | Data Tier |
| Scripts and Customizations | BIHOST1 and BIHOST2 | Application Tier |
| Deployment Plan home (DEPLOY_PLAN_HOME) | BIHOST1 and BIHOST2 | Application Tier |

***Table 13-2 (Cont.) Run-Time Artifacts to Back Up in the Oracle Business Intelligence Enterprise Deployment***

| Type | Host | Tier |
|---|---|---|
| Singleton Data Directory (SDD) | BIHOST1 and BIHOST2 | Application Tier |

# 14

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

The following topics describe Oracle WebLogic Server Whole Server migration and Oracle WebLogic Server Automatic Service Migration. The also explain how these features can be used in an Oracle Fusion Middleware enterprise topology.

**See Also:**



## 14.1 About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment

Oracle WebLogic Server provides a migration framework that is 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.

**See Also:**

Understanding Which Products and Components Require Whole Server Migration and Service Migration

### 14.1.1 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 of its services, on a different physical machine. When a failure occurs in a server that is part of a cluster that 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 need to use a a floating IP as listen address and the required resources (transactions logs and JMS persistent stores) must be available on the candidate machines.

  For more information, 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.

  For more information, see "Understanding the Service Migration Framework" in *Administering Clusters for Oracle WebLogic Server*.

### 14.1.2 Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment

When a server or service is started in another system, the required resources (such as services data and logs) must be available to both the original system and to the failover system; otherwise, the service cannot resume the same operations successfully on the failover system.

For this reason, both whole server and service migration require that all members of the cluster have access to the same transaction and JMS persistent stores (whether the persistent store is file-based or database-based).

This is another reason why shared storage is important in an enterprise deployment. When you properly configure shared storage, you ensure that in the event of a manual failover (Administration Server failover) or an automatic failover (whole server migration or service migration), both the original machine and the failover machine can access the same file store with no change in service.

In the case of an automatic service migration, when a pinned service needs to be resumed, the JMS and JTA logs that it was using before failover need to be accessible.

In addition to shared storage, Whole Server Migration requires the procurement and assignment of a virtual IP address (VIP). When a Managed Server fails over to another machine, the VIP is automatically reassigned to the new machine.

Note that service migration does not require a VIP.

### 14.1.3 Understanding Which Products and Components Require Whole Server Migration and Service Migration

The following table summarizes the products and components you should configure for whole server migration and the products you should configure for Automatic 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 Business Intelligence Publisher | YES | NO |

## 14.2 Creating a GridLink Data Source for Leasing

Both 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).

For an enterprise deployment, you should use create a GridLink data source:

1. Log in to the Oracle WebLogic Server Administration Console.

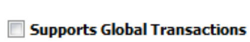
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.

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:

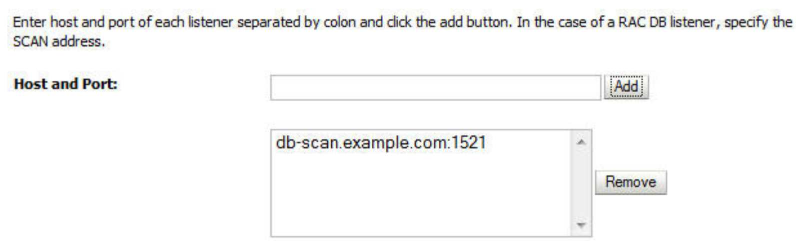
   - **Service Name**: Enter the service name of the database with lowercase characters. For a GridLink data source, you must enter the Oracle RAC service name. For example:

     ```
     biedg.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. For information about configuring multi data sources see Using Multi Data Sources with Oracle RAC.

   - **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=biedg.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** and 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**.

## 14.3 Configuring Whole Server Migration for an Enterprise Deployment

After you have prepared your domain for whole server migration or automatic service migration, you can then configure Whole Server Migration for specific Managed Servers within a cluster. See the following topics for more information.

**See Also:**

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

### 14.3.1 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 or not to use a node's MAC address when sending ARP packets, that is, whether or not to use the `-b` flag in the arping command.

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
...
```

## 14.3.2 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.

**See Also:**

Setting the PATH Environment Variable for the wlsifconfig.sh Script

Granting Privileges to the wlsifconfig.sh Script

### 14.3.2.1 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` |

| File | Directory Location |
|---|---|
| `nodemanager.domains` | *MSERVER_HOME*/nodemanager |

#### 14.3.2.2 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
```

### 14.3.3 Configuring Server Migration Targets

To configure migration in a cluster:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure window, expand **Environment** and select **Clusters**. The Summary of Clusters page appears.
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. Verify that **Database** is the selected leasing mechanism.

   If necessary, select **Database** as the leasing mechanism.
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. Select the Leasing data source that you created in Creating a GridLink Data Source for Leasing.
9. Click **Save**.
10. 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.

   For information on targeting applications and resources, see Using Multi Data Sources with Oracle RAC.

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 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.

11. Click **Activate Changes**.

12. Restart the Administration Server and the servers for which server migration has been configured.

### 14.3.4 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:

   ```
   ps -ef | grep WLS_BI1
   ```

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:

`MSERVER_HOME/servers/WLS_BI1/data/nodemanager/startup.properties`

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 using 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.

---

## 14.4 Configuring Automatic Service Migration in an Enterprise Deployment

To configure automatic service migration for specific services in an enterprise deployment, refer to the topics in this section.

---

**Note:**

Oracle Business Intelligence currently does not support automatic service migration. The information is included here for users who are deploying other Fusion Middleware products that do support automatic service migration.

---

**See Also:**

Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster

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

Restarting the Managed Servers and Validating Automatic Service Migration

Failing Back Services After Automatic Service Migration

### 14.4.1 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.

### 14.4.2 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 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**.

### 14.4.3 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 will be 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 are using 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. With this policy, the services hosted by the migratable target will start only if the migratable target's User Preferred Server (UPS) is started.

For more information, see Policies for Manual and Automatic Service Migration in *Administering Clusters for Oracle WebLogic Server*.

### 14.4.4 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 haven’t 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.

   For more information, see About Selecting a Service Migration Policy.
6. Click **Save**.
7. Repeat steps 2 through 6 for each of the additional Managed Servers in the cluster.
8. Activate the changes.
9. Restart the Managed Servers in the cluster.

### 14.4.5 Restarting the Managed Servers and Validating Automatic Service Migration

After you configure automatic service migration for your cluster and Managed Servers, validate the configuration, as follows:

1. If you haven't already, log in to the Administration Console.
2. In the Domain Structure pane, select **Environment**, then **Clusters**, and restart the cluster you just configured for automatic service migration.
3. In the Domain Structure pane, expand **Environment**, and then expand **Clusters**.
4. Click **Migratable Targets**.
5. Click the **Control** tab.

   The console displays a list of migratable targets and their current hosting server.
6. In the Migratable Targets table, select a row for the one of the migratable targets.
7. Note the value in the **Current Hosting Server** column.
8. Use the operating system command line to stop the the first Managed Server.

   Use the following command to kill 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 that after you kill the process, the Managed Server might be configured to start automatically after you initially kill the process. In this case, you must kill the second process using the `kill -9` command again.
9. 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 will be 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 so attempting to restart (restart count = 1)>.
```

10. 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".

11. 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 destintations (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. Click the destination in the **Summary of Resources** table. destination->Select monitoring and get the Messages Total and Messages Pending Counts

d. Select the Monitoring tab, and review the **Messages Total** and **Messages Pending** values in the **Destinations** table.

---

### 14.4.6 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 and the configuration edit lock is released. In the Migratable Targets table, the Status of Last Migration column indicates whether the requested migration has succeeded or failed.

4. To migrate a specific migratable target, on the Summary of Migratable Targets page:

   a. Select the migratable target to migrate.

   b. Click the **Control** tab.

   c. Reselect the migratable target to migrate.

   d. Click **Migrate**.

   e. Use the New hosting server drop-down to select a new server for the migratable target.

   f. Click **OK**.

# 15

# Configuring Single Sign-On for an Enterprise Deployment

This chapter describes how to configure the Oracle HTTP Server WebGate to enable single sign-on with Oracle Access Manager.

**See Also:**

## 15.1 About Oracle HTTP Server Webgate

Oracle HTTP Server 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 WebGate software is installed as part of the Oracle HTTP Server 12*c* software installation.

For more extensive information about WebGates, see "Registering and Managing OAM 11g Agents" in the *Adminstrator's Guide for Oracle Access Management*.

## 15.2 General Prerequisites for Configuring Oracle HTTP Server 12*c* Webgate

Before you can configure Oracle HTTP Server 12*c* WebGate, you must have installed and configured a certified version of Oracle Access Manager.

At the time this document was published, the supported versions of Oracle Access Manager were 11*g* Relase 2 (11.1.2.2) and 11*g* Release 2 (11.1.2.3). For the most up-to-date information, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

---

**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.

---

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.

## 15.3 Enterprise Deployment Prerequisites for Configuring OHS 12*c* Webgate

When you are configuring Oracle HTTP Server Webgate to enable Single Sign-On for an enterprise deployment, consider the prerequisites mentioned in this section.

- Oracle recommends that you deploy Oracle Access Manager as part a highly available, secure, production environment. For more information about deploying Oracle Access Manager in an enterprise environment, see the Enterprise Deployment Guide for your version of Oracle Identity and Access Mangement.
- To enable single sign-on for the WebLogic Server Administration Console and the Oracle Enterprise Manager Fusion Middleware Control, you must add a central LDAP-provisioned administration user to the directory service that Oracle Access Manager is using (for example, Oracle Internet Directory or Oracle Unified Directory). For more information about the required user and groups to add to the LDAP directory, follow the instructions in Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group.

## 15.4 Configuring Oracle HTTP Server 12*c* WebGate for an Enterprise Deployment

Perform the following steps 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 *OHS_ORACLE_HOME* and *OHS_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 OHS_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 OHS_CONFIG_DIR -oh OHS_ORACLE_HOME
   ```

4. Verify that a `webgate` directory and subdirectories was created by the `deployWebGateInstance` command:

   ```
   ls -lart OHS_CONFIG_DIR/webgate/
   total 6
   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 *`OHS_ORACLE_HOME`*`/lib` directory path:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:OHS_ORACLE_HOME/lib
   ```

6. Change directory to the following directory

   ```
   OHS_ORACLE_HOME/webgate/ohs/tools/setup/InstallTools
   ```

7. Run the following command from the `InstallTools` directory.

   ```
   ./EditHttpConf -w OHS_CONFIG_DIR -oh OHS_ORACLE_HOME -o
   output_file_name
   ```

   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 *`output_file`* argument to the command.

# 15.5 Registering the Oracle HTTP Server 12*c* WebGate with Oracle Access Manager

You can register the WebGate agent with Oracle Access Manager by using the Oracle Access Manager Administration console.

For more information, see "Registering an OAM Agent Using the Console" in the *Oracle Fusion Middleware Administrator's Guide for Oracle Access Management*.

For more information, see the following topics:

**See Also:**

Locating and Preparing the RREG Tool

About RREG In-Band and Out-of-Band Mode

Updating the Standard Properties in the OAM11gRequest.xml File

Updating the Protected, Public, and Excluded Resources for an Enterprise Deployment

Running the RREG Tool

Files and Artifacts Generated by RREG

Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location

Restarting the Oracle HTTP Server Instance

## 15.5.1 Locating and Preparing the RREG Tool

To set up the RREG tool, complete the following steps:

1. Log in to one of the Oracle Access Manager hosts in the Application tier.

2. Change directory to the following directory in the Oracle Access Manager Oracle home:

   `OAM_ORACLE_HOME/oam/server/rreg/client`

   In this example, *OAM_ORACLE_HOME* refers to the Oracle home on the system where the Oracle Access Manager software was installed.

   > **Note:** If you do not have privileges or access to the Oracle Access Manager server, then you can use out-of-band mode to generate the required files and register the WebGate with Oracle Access Manager. For more information, see About RREG In-Band and Out-of-Band Mode.

3. Open the `oamreg.sh` file and set the following environment variables in the file, as follows:

   - Set `OAM_REG_HOME` to the absolute path to the directory in which you extracted the contents of RREG archive.

     Set `JDK_HOME` to the absolute path of the directory in which a supported JDK is installed on your machine.

### 15.5.2 About RREG In-Band and Out-of-Band Mode

You can run the RREG Tool in one of two modes: in-band and out-of-band.

Use **in-band** mode when you have the privileges to access the Oracle Access Manager server and run the RREG tool yourself from the Oracle Access Manager Oracle home. You can then copy the generated artifacts and files to the Web server configuration directory after you run the RREG Tool.

Use **out-of-band** mode if you do *not* have privileges or access to the Oracle Access Manager server. For example, in some organizations, only the Oracle Access Manager server administrators have privileges access the server directories and perform administration tasks on the server. In out-of-band mode, the process can work as follows:

1. The Oracle Access Manager server administrator provides you with a copy of the RREG archive file (RREG.tar.gz), which the server administrator can find in the location described in Locating and Preparing the RREG Tool.

2. Untar the `RREG.tar.gz` file that was provided to you by the server administrator.

   For example:

   ```
   gunzip RREG.tar.gz
   ```

   ```
   tar -xvf RREG.tar
   ```

   After you unpack the RREG archive, you can find the tool for registering the agent in the following location:

   *`RREG_HOME`*`/bin/oamreg.sh`

   In this example, *`RREG_Home`* is the directory in which you extracted the contents of RREG archive.

3. Use the instructions in Updating the Standard Properties in the OAM11gRequest.xml File to update the `OAM11GRequest.xml` file, and send the completed `OAM11GRequest.xml` file to the Oracle Access Manager server administrator.

4. The Oracle Access Manager server administrator then uses the instructions in Running the RREG Tool in Out-Of-Band Mode to run the RREG Tool and generate the *`AgentID`*`_response.xml` file.

5. The Oracle Access Manager server administrator sends the *`AgentID`*`_response.xml` file to you.

6. Use the instructions in Running the RREG Tool in Out-Of-Band Mode to run the RREG Tool with the *`AgentID`*`_response.xml` file and generate the required artifacts and files on the client system.

### 15.5.3 Updating the Standard Properties in the OAM11gRequest.xml File

Before you can register the Webgate agent with Oracle Access Manager, you must update some required properties in the `OAM11gRequest.xml` file.

**Note:**

If you plan to use the default values for most of the parameters in the provided XML file, then you can use the shorter version (`OAM11gRequest_short.xml`, in which all non-listed fields will take a default value.

To perform this task:

1. If you are using in-band mode, then change directory to the following location in the directory:

   ```
   OAM_ORACLE_HOME/oam/server/rreg/client
   ```

   If you are using out-of-band mode, then change directory to the location where you unpacked the RREG archive.

2. Make a copy of the `OAM11gRequest.xml` file template.

3. Review the properties listed in the file, and then update your copy of the OAM11gRequest.xml file to make sure the properties reference the host names and other values specific to your environment.

| OAM11gRequest.xml Property | Set to... |
|---|---|
| `serverAddress` | The host and the port of the Administration Server for the Oracle Access Manager domain. |
| `agentName` | Any custom name for the agent. Typically, you use a name that identifies the Fusion Middleware product you are configuring for single sign-on. |
| `applicationDomain` | A value that identifies the Web tier host and the FMW component you are configuring for single sign-on. |
| `security` | The security mode of the Oracle Access Manager server, which can be open, simple, or certificate mode.<br>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.<br>In most cases, avoid using open mode, because in open mode, traffic to and from the Oracle Access Manager server is not encrypted.<br>For more information using certificate mode or about Oracle Access Manager supported security modes in general, see Securing Communication Between OAM Servers and WebGatesin the *Administrator's Guide for Oracle Access Management*. |
| `cachePragmaHeader` | private |
| `cacheControlHeader` | private |
| `ipValidation` | 0<br>`<ipValidation>0<ipValidation>` |

| | |
|---|---|
| `ipValidationExceptions` | The IP address of the front-end load balancer. For example:<br>`<ipValidationExceptions>`<br>`    <ipAddress>130.35.165.42</ipAddress>`<br>`</ipValidation>` |
| `agentBaseUrl` | The host and the port of the machine on which Oracle HTTP Server 12*c* WebGate is installed. |

### 15.5.4 Updating the Protected, Public, and Excluded Resources for an Enterprise Deployment

When you set up an Oracle Fusion Middleware environment for single sign-on, you identify a set of URLs that you want Oracle Access Manager to protect with single sign-on. You identify these using specific sections of the `OAM11gRequest.xml` file. To identify the URLs:

1. If you haven't already opened `OAM11gRequest.xml` file for editing, locate and open the file in a text editor.

   For more information, see the following:

   - Locating and Preparing the RREG Tool
   - Updating the Standard Properties in the OAM11gRequest.xml File

2. Remove the sample entries from the file, and then enter the list of protected, public, and excluded resources in the appropriate sections of the file, as shown in the following example.

   **Note:**

   If you are using Oracle Access Manager 11*g* Release 2 (11.1.2.2) or later, then note that the entries with the wildcard syntax ("`.../*`") are included for backward compatibility with previous versions of Oracle Access Manager.

```
<protectedResourcesList>
        <resource>/analytics/saw.dll</resource>
        <resource>/bicontent</resource>
        <resource>/xmlpserver</resource>
        <resource>/mapviewer</resource>
        <resource>/bicomposer</resource>
        <resource>/bisearch</resource>
        <resource>/em</resource>
        <resource>/em/…/*</resource>
        <resource>/console</resource>
        <resource>/console/…/*</resource>
        <resource>/mobile</resource>
        <resource>/mobile/.../*</resource>
        <resource>/va</resource>
</protectedResourcesList>
<publicResourcesList>
        <resource>/analytics</resource>
        <resource>/analytics-ws/saw.dll</resource>
```

```
        <resource>/aps</resource>
        <resource>/aps/JAPI</resource>
        <resource>/aps/Essbase</resource>
</publicResourcesList>
<excludedResourcesList>
        <resource>/biservices</resource>
        <resource>/analytics-bi-adf</resource>
        <resource>/xmlpserver/Guest</resource>
        <resource>/xmlpserver/ReportTemplateService.xls</resource>
        <resource>/xmlpserver/report_service</resource>
        <resource>/xmlpserver/services</resource>
        <resource>/analytics/saw.dll/wsdl</resource>
        <resource>/analytics-ws</resource>
        <resource>/ws/.../*</resource>
        <resource>/wsm-pm</resource>
        <resource>/wsm-pm/.../*</resource>
</excludedResourcesList>
```

3. Save and close the `OAM11GRequest.xml` file.

## 15.5.5 Running the RREG Tool

The following topics provide information about running the RREG tool to register your Oracle HTTP Server Webgate with Oracle Access Manager.

**See Also:**

Running the RREG Tool in In-Band Mode

Running the RREG Tool in Out-Of-Band Mode

### 15.5.5.1 Running the RREG Tool in In-Band Mode

To run the RREG Tool in in-band mode:

1. Change directory the RREG home directory.

   If you are using in-band mode, the RREG directory is inside the Oracle Access Manager Oracle home:

   ```
   OAM_ORACLE_HOME/oam/server/rreg/client
   ```

   If you are using out-of-band mode, then the RREG home directory is the location where you unpacked the RREG archive.

2. Change directory to the bin directory inside the RREG home directory:

   ```
   cd RREG_HOME/bin/
   ```

3. Set the permissions of the `oamreg.sh` command so you can execute the file:

   ```
   chmod +x oamreg.sh
   ```

4. Run the following command:

   ```
   ./oamreg.sh inband input/OAM11GRequest.xml
   ```

In this example:

- It is assumed the edited `OAM11GRequest.xml` file is located in the *`RREG_HOME/`*`input` directory.
- The output from this command will be saved to the following directory:

```
RREG_HOME/output/
```

The following example shows a sample RREG session:

```
Welcome to OAM Remote Registration Tool!
Parameters passed to the registration tool are:
Mode: inband
Filename: /u01/oracle/products/fmw/iam_home/oam/server/rreg/client/rreg/input/
OAM11GWCCDomainRequest.xml
Enter admin username:weblogic_idm
Username: weblogic_idm
Enter admin password:
Do you want to enter a Webgate password?(y/n):
n
Do you want to import an URIs file?(y/n):
n

----------------------------------------
Request summary:
OAM11G Agent Name:WCC1221_EDG_AGENT
URL String:null
Registering in Mode:inband
Your registration request is being sent to the Admin server at: http://
host1.example.com:7001
----------------------------------------

Jul 08, 2015 7:18:13 PM oracle.security.jps.util.JpsUtil disableAudit
INFO: JpsUtil: isAuditDisabled set to true
Jul 08, 2015 7:18:14 PM oracle.security.jps.util.JpsUtil disableAudit
INFO: JpsUtil: isAuditDisabled set to true
Inband registration process completed successfully! Output artifacts are created in
the output folder.
```

#### 15.5.5.2 Running the RREG Tool in Out-Of-Band Mode

To run the RREG Tool in out-of-band mode on the Oracle Access Manager server, the Oracle Access Manager server administrator uses the following command:

```
RREG_HOME/bin/oamreg.sh outofband input/OAM11GRequest.xml
```

In this example:

- Replace *RREG_HOME* with the location where the RREG archive file was unpacked on the server.
- The edited `OAM11GRequest.xml` file is located in the *`RREG_HOME/`*`input` directory.
- The RREG Tool saves the output from this command (the *`AgentID`*`_response.xml` file) to the following directory:

  ```
  RREG_HOME/output/
  ```

  The Oracle Access Manager server administrator can then send the *`AgentID`*`_response.xml` to the user who provided the `OAM11GRequest.xml` file.

To run the RREG Tool in out-of-band mode on the Web server client machine, use the following command:

```
RREG_HOME/bin/oamreg.sh outofband input/AgentID_response.xml
```

In this example:

- Replace *RREG_HOME* with the location where you unpacked the RREG archive file on the client system.
- The `AgentID_response.xml` file, which was provided by the Oracle Access Manager server administrator, is located in the *RREG_HOME*/input directory.
- The RREG Tool saves the output from this command (the artifacts and files required to register the Webgate software) to the following directory on the client machine:

  *RREG_HOME*/output/

### 15.5.6 Files and Artifacts Generated by RREG

The files that get generated by the RREG Tool vary, depending on the security level you are using for communications between the WebGate and the Oracle Access Manager server. For more information about the supported security levels, see Securing Communication Between OAM Servers and WebGates in the *Administrator's Guide for Oracle Access Management*.

Note that in this topic any references to *RREG_HOME* should be replaced with the path to the directory where you ran the RREG tool. This is typically the following directory on the Oracle Access Manager server, or (if you are using out-of-band mode) the directory where you unpacked the RREG archive:

*OAM_ORACLE_HOME*/oam/server/rreg/client

The following table lists the artifacts that are always generated by the RREG Tool, regardless of the Oracle Access Manager security level.

| File | Location |
|---|---|
| `cwallet.sso` | *RREG_HOME*/output/Agent_ID/ |
| `ObAccessClient.xml` | *RREG_HOME*/output/Agent_ID/ |

The following table lists the additional files that are created if you are using the SIMPLE or CERT security level for Oracle Access Manager:

| File | Location |
|---|---|
| `aaa_key.pem` | *RREG_HOME*/output/Agent_ID/ |
| `aaa_cert.pem` | *RREG_HOME*/output/Agent_ID/ |
| `password.xml` | *RREG_HOME*/output/Agent_ID/ |

Note that the `password.xml` file contains the obfuscated global passphrase to encrypt the private key used in SSL. This passphrase can be different than the passphrase used on the server.

You can use the files generated by RREG to generate a certificate request and get it signed by a third-party Certification Authority. To install an existing certificate, you must use the existing `aaa_cert.pem` and `aaa_chain.pem` files along with `password.xml` and `aaa_key.pem`.

### 15.5.7 Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location

After the RREG Tool generates the required artifacts, manually copy the artifacts from the `RREG_Home/output/agent_ID` directory to the Oracle HTTP Server configuration directory on the Web tier host.

The location of the files in the Oracle HTTP Server configuration directory depends upon 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 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 using certificate mode or about Oracle Access Manager supported security modes in general, see Securing Communication Between OAM Servers and WebGatesin the *Administrator's Guide for Oracle Access Management*.

---

| File | Location When Using OPEN Mode | Location When Using SIMPLE Mode | Location When Using CERT Mode |
|---|---|---|---|
| `wallet/cwallet.sso` | *OHS_CONFIG_DIR/* `webgate/config/wallet` | *OHS_CONFIG_DIR/* `webgate/config/wallet/` | *OHS_CONFIG_DIR/* `webgate/config/wallet/` |
| `ObAccessClient.xml` | *OHS_CONFIG_DIR/* `webgate/config` | *OHS_CONFIG_DIR/* `webgate/config/` | *OHS_CONFIG_DIR/* `webgate/config/` |
| `password.xml` | N/A | *OHS_CONFIG_DIR/* `webgate/config/` | *OHS_CONFIG_DIR/* `webgate/config/` |
| `aaa_key.pem` | N/A | *OHS_CONFIG_DIR/* `webgate/config/simple/` | *OHS_CONFIG_DIR/* `webgate/config/` |
| `aaa_cert.pem` | N/A | *OHS_CONFIG_DIR/* `webgate/config/simple/` | *OHS_CONFIG_DIR/* `webgate/config/` |

---

**Note:** When you copy `ObAccessClient.xml` to a new directory, you will need to delete the old `ObAccessClient.xml` file from the first Oracle HTTP Server location cache location on WEBHOST1:

```
OHS_DOMAIN_HOME/servers/ohs1/cache/config/
```

And you must perform the similar step for the second Oracle HTTP Server instance on WEBHOST2:

```
OHS_DOMAIN_HOME/servers/ohs2/cache/config/
```

---

### 15.5.8 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. For more information, see "Restarting Oracle HTTP Server Instances by Using Fusion Middleware Control" in *Administrator's Guide for Oracle HTTP Server*.

## 15.6 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 continuing with this section.

**See Also:**

Backing Up Configuration Files

Setting Up the Oracle Access Manager Identity Assertion Provider

Setting the Order of Providers

### 15.6.1 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
```

### 15.6.2 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. Select both the **ObSSOCookie** and **OAM_REMOTE_USER** options under Chosen types.
11. Click **Save** to save the settings.
12. Click **Activate Changes** to propagate the changes.
13. Restart the Administration Server and Managed Servers.

## 15.6.3 Setting the Order of Providers

Set the order of identity assertion and authentication providers in the WebLogic Server Administration Console.

To 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. Click **Security Realms** in the left navigation bar.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. Reorder the Oracle Access Manager identity assertion provider, the LDAP authentication provider, and the default authentication provider by ensuring that the control flag for each provider is set as follows:
   - Oracle Access Manager identity assertion provider: REQUIRED
   - LDAP authentication provider: SUFFICIENT
   - Default authentication provider (`DefaultAuthenticator`): SUFFICIENT
7. Click **OK**.
8. Click **Activate Changes** to propagate the changes.
9. Restart the Administration Server, Managed Servers, and system components.

## 15.7 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:

```
DOMAIN_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 in SSO actions in Oracle Access Manager, complete the following steps:

1. Change directory to the following directory:

   ```
   cd ORACLE_COMMON_HOME/common/bin
   ```

2. Start the WebLogic Server Scripting Tool (WLST):

   ```
   ./wlst.sh
   ```

3. Connect to the Administration Server, using the following WLST command:

   ```
   connect('admin_user','admin_password','admin_url')
   ```

   For example:

   ```
   connect('weblogic','mypassword','t3://ADMINVHN:7001')
   ```

4. Execute the `addOAMSSOProvider` command, as follows:

   ```
   addOAMSSOProvider(loginuri="/${app.context}/adfAuthentication", logouturi="/
   oamsso/logout.html")
   ```

   The following table defines the expected value for each argument in the addOAMSSOProvider command.

| Argument | Definition |
| --- | --- |

| | |
|---|---|
| loginuri | Specifies the URI of the login page<br>Note: For ADF security enabled applications, "<*context-root*/adfAuthentication" should be provided for the 'loginuri' parameter.<br>For example:<br>`/${app.context}/adfAuthentication`<br>Here is the flow:<br>**a.** User accesses a resource that has been protected by authorization policies in OPSS, fox example.<br>**b.** If the user is not yet authenticated, ADF redirects the user to the URI configured in 'loginuri'.<br>**c.** Access Manager, should have a policy to protect the value in 'loginuri': for example, "/<context-root/adfAuthentication.<br>**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<br>Notes:<br>• For ADF security enabled applications, `logouturi` should be configured based on logout guidelines in "Configuring Centralized Logout for Sessions Involving 11g WebGates" in the *Administrator's Guide for Oracle Access Management*.<br>• When using WebGate 11*g*, the value of the `logouturi` should be sought from the 11g WebGate Administrator.<br>• When using WebGate 10g, the value of `logouturi` should be `/oamsso/logout.html`. |
| autologinuri | Specifies the URI of the autologin page. This is an optional parameter. |

5. Disconnect from the Administration Server:

```
disconnect()
```

6. Restart the Administration Server and all the Managed Servers.

## 15.8 Configuring Single Sign-On for Applications

This section describes how to enable single sign-on (SSO) for BI applications.

It includes the following topics.

**See Also:**

Enabling Single Sign-On and Oracle Access Manager for Oracle BI EE

Enabling Single Sign-On and Oracle Access Manager for Oracle BI Publisher

### 15.8.1 Enabling Single Sign-On and Oracle Access Manager for Oracle BI EE

Perform the following steps to enable single sign-on (SSO) and Oracle Access Manager for Oracle Business Intelligence Enterprise Edition (BI EE):

1. Start WLST:

   ```
   cd ORACLE_HOME/oracle_common/common/bin
   ./wlst.sh
   ```

2. Open the BI Administration Server domain for updating:

   ```
   wls:/offline> readDomain('ASERVER_HOME')
   ```

   In this example, replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.

3. Run the following command to enable SSO in Oracle BI EE and configure the logout information for the Oracle BI Presentation Services processes:

   ```
   wls:/offline/bi_domain> enableBISingleSignOn('ASERVER_HOME','http://
   oam_host:oam_port/oamsso/logout.html')
   ```

   In this example:

   - Replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.
   - `http://oam_host:oam_port/oamsso/logout.html` is the SSO provider (Oracle Access Manager) logoff URL.

4. Update and save the domain:

   ```
   wls:/offline/bi_domain> updateDomain()
   ```

5. Close the domain:

   ```
   wls:/offline/bi_domain> closeDomain()
   ```

6. Exit WLST:

   ```
   wls:/offline> exit()
   ```

7. Restart the Administration Server, Managed Servers, and system components.

### 15.8.2 Enabling Single Sign-On and Oracle Access Manager for Oracle BI Publisher

Perform the following steps to enable single sign-on (SSO) and Oracle Access Manager for Oracle BI Publisher:

1. Log in to BI Publisher using one of the following URLs:

   - http://BIHOST1VHN1:7003/xmlpserver
   - http://BIHOST2VHN1:7003/xmlpserver

   You will be redirected to:

   ```
   http://bi.example.com/xmlpserver
   ```

2. In BI Publisher, go to the **Administration** > **Security Configuration** page to enable SSO.
3. On the Security Configuration page, provide the following information in the Single Sign-On section:
   a. Select **Use Single Sign-On**.
   b. For **Single Sign-On Type**, select **Oracle Access Manager**.
   c. For **Single Sign-Off URL**, enter a URL of the following format:

      `http://oam_host:oam_port/oamsso/logout.html`

   d. For **User Name Parameter**, enter `OAM_REMOTE_USER`.
4. Click **Apply**.
5. Restart the **bipublisher** application from the WebLogic Administration Console.

   For information on how to restart the **bipublisher** application, see Using Oracle WebLogic Server Administration Console to Start and Stop Java Components in the *System Administrator's Guide for Oracle Business Intelligence Enterprise Edition*.

# A

# Using Multi Data Sources with Oracle RAC

Oracle recommends using GridLink data sources when developing new Oracle RAC applications. However, if you are using legacy applications and databases that do not support GridLink data sources, refer to the information in this appendix.

This appendix provides information about multi data sources and Oracle RAC and procedure for configuring multi data sources for an Enterprise Deployment.

**See Also:**



## A.1 About Multi Data Sources and Oracle RAC

A multi data source provides an ordered list of data sources to use to satisfy connection requests.

Normally, every connection request to this kind of multi data source is served by the first data source in the list. If a database connection test fails and the connection cannot be replaced, or if the data source is suspended, a connection is sought sequentially from the next data source on the list.

For more information about configuring Multi Data Sources with Oracle RAC, see "Using Multi Data Sources with Oracle RAC" in the *Oracle Fusion Middleware Configuring and Managing JDBC Data Sources for Oracle WebLogic Server*.

## A.2 Typical Procedure for Configuring Multi Data Sources for an Enterprise Deployment

You configure data sources when you configure a domain. If you want to use Multi Data Sources instead of GridLink data sources, replace the GridLink instructions with the instructions provided in this section.

For example, when you are configuring the initial Administration domain for an Enterprise Deployment reference topology, you use the configuration wizard to define the characteristics of the domain, as well as the data sources.

The procedures for configuring the topologies in this Enterprise Deployment Guide include specific instructions for defining GridLink data sources with Oracle RAC. If you want to use Multi Data Sources instead of GridLink data sources, replace the GridLink instructions with the following:

1. In the Configure JDBC Component Schema screen:

   a. Select the appropriate schemas.

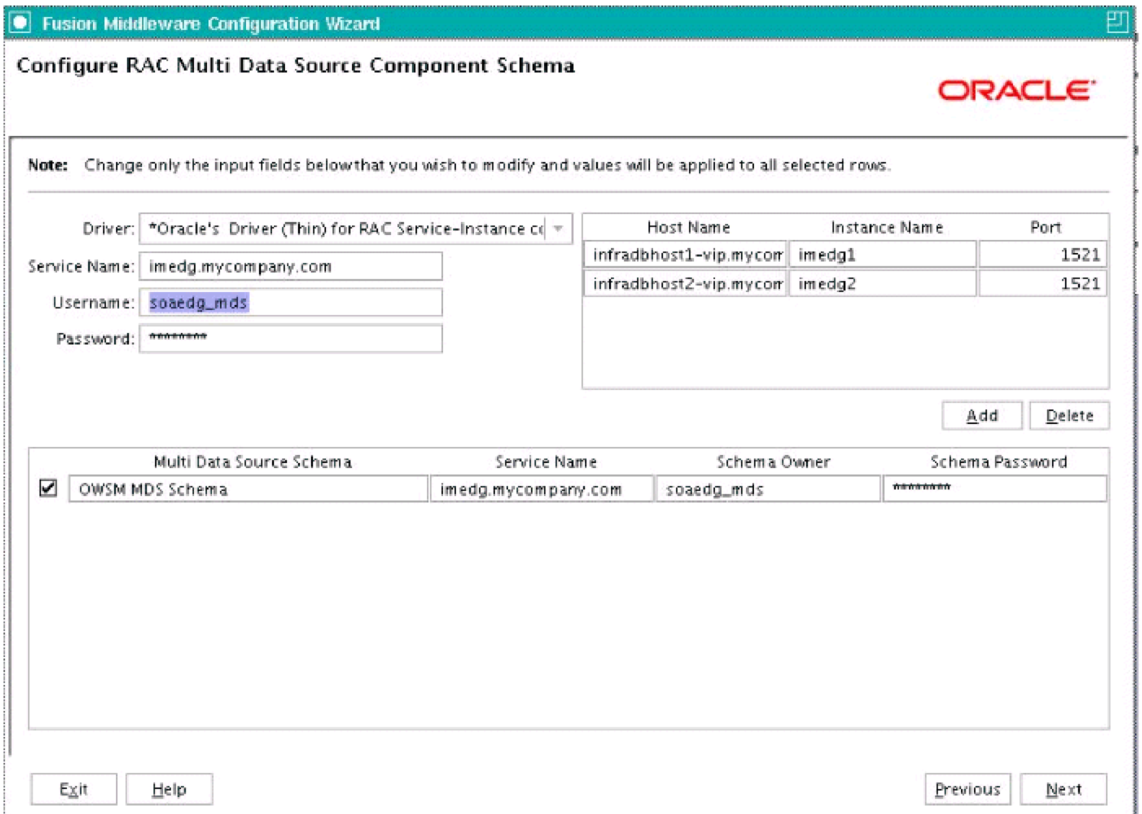
   b. For the RAC configuration for component schemas, **Convert to RAC multi data source**.

   c. Ensure that the following data source appears on the screen with the schema prefix when you ran the Repository Creation Utility.

   d. Click **Next**.

2. The Configure RAC Multi Data Sources Component Schema screen appears (Figure A-1).

***Figure A-1 Configure RAC Multi Data Source Component Schema Screen***




In this screen, do the following:

a. Enter values for the following fields, specifying the connect information for the Oracle RAC database that was seeded with RCU.

   - **Driver:** Select **Oracle driver (Thin) for RAC Service-Instance connections, Versions:10, 11**.
   - **Service Name:** Enter the service name of the database.
   - **Username:** Enter the complete user name (including the prefix) for the schemas.
   - **Password:** Enter the password to use to access the schemas.

b. Enter the host name, instance name, and port.

c. Click **Add**.

d. Repeat this for each Oracle RAC instance.

e. Click **Next**.

3. In the Test JDBC Data Sources screen, the connections are tested automatically. The **Status** column displays the results. Ensure that all connections were successful. If not, click **Previous** to return to the previous screen and correct your entries.

   Click **Next** when all the connections are successful.

# Index