Oracle® Fusion Middleware

Enterprise Deployment Guide for Oracle Business Intelligence

12*c* (12.2.1.3.0)
E95620-02
July 2019

ORACLE

Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence, 12*c* (12.2.1.3.0)

E95620-02

Copyright © 2015, 2019, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience xii
Documentation Accessibility xii
Conventions xii

Part I Understanding an Enterprise Deployment

1 Enterprise Deployment Overview

About the Enterprise Deployment Guide 1-1
When to Use the Enterprise Deployment Guide 1-2

2 About a Typical Enterprise Deployment

Diagram of a Typical Enterprise Deployment 2-1
About the Typical Enterprise Deployment Topology Diagram 2-3
- Understanding the Firewalls and Zones of a Typical Enterprise Deployment 2-3
- Understanding the Elements of a Typical Enterprise Deployment Topology 2-4
- Receiving Requests Through Hardware Load Balancer 2-4
  - Purpose of the Hardware Load Balancer (LBR) 2-4
  - Summary of the Typical Load Balancer Virtual Server Names 2-7
  - HTTPS Versus HTTP Requests to the External Virtual Server Name 2-7
- Understanding the Web Tier 2-7
  - Benefits of Using a Web Tier to Route Requests 2-8
  - Alternatives to Using a Web Tier 2-9
  - Configuration of Oracle HTTP Server in the Web Tier 2-9
  - About Mod_WL_OHS 2-9
- Understanding the Application Tier 2-9
  - Configuration of the Administration Server and Managed Servers Domain Directories 2-10
  - Using Oracle Web Services Manager in the Application Tier 2-11
  - Best Practices and Variations on the Configuration of the Clusters and Hosts on the Application Tier 2-11

About the Node Manager Configuration in a Typical Enterprise Deployment 2-12
About Using Unicast for Communications within the Application Tier 2-13
Understanding OPSS and Requests to the Authentication and Authorization Stores 2-14
About the Data Tier 2-14

3 Understanding the Business Intelligence Enterprise Deployment Topology

Diagram of the Primary Business Intelligence Enterprise Topology 3-2
Understanding the Primary Business Intelligence Topology Diagram 3-3
Summary of Business Intelligence Load Balancer Virtual Server Names 3-3
Summary of the Managed Servers and Cluster on the Business Intelligence Application Tier 3-4
Flow Charts and Roadmaps for Implementing the Primary Business Intelligence Enterprise Topologies 3-4
Flow Chart of the Steps to Install and Configure the Primary Business Intelligence Enterprise Topologies 3-4
Roadmap Table for Planning and Preparing for an Enterprise Deployment 3-5
Roadmap Table for Configuring the Business Intelligence Enterprise Topology 3-6

Part II Preparing for an Enterprise Deployment

4 Using the Enterprise Deployment Workbook

Introduction to the Enterprise Deployment Workbook 4-1
Typical Use Case for Using the Workbook 4-2
Using the Oracle Business Intelligence Enterprise Deployment Workbook 4-2
Locating the Oracle Business Intelligence Enterprise Deployment Workbook 4-2
Understanding the Contents of the Oracle Business Intelligence Enterprise Deployment Workbook 4-2
Using the Start Tab 4-3
Using the Hardware - Host Computers Tab 4-3
Using the Network - Virtual Hosts & Ports Tab 4-4
Using the Storage - Directory Variables Tab 4-4
Using the Database - Connection Details Tab 4-5
Who Should Use the Enterprise Deployment Workbook? 4-5

5 Procuring Resources for an Enterprise Deployment

Hardware and Software Requirements for the Enterprise Deployment Topology 5-1
Hardware Load Balancer Requirements 5-1

Host Computer Hardware Requirements 5-2
General Considerations for Enterprise Deployment Host Computers 5-3
Reviewing the Oracle Fusion Middleware System Requirements 5-3
Typical Memory, File Descriptors, and Processes Required for an Enterprise Deployment 5-4
Typical Disk Space Requirements for an Enterprise Deployment 5-4
Operating System Requirements for an Enterprise Deployment Topology 5-5
Reserving the Required IP Addresses for an Enterprise Deployment 5-5
What is a Virtual IP (VIP) Address? 5-6
Why Use Virtual Host Names and Virtual IP Addresses? 5-6
Physical and Virtual IP Addresses Required by the Enterprise Topology 5-7
Identifying and Obtaining Software Distributions for an Enterprise Deployment 5-8

6 Preparing the Load Balancer and Firewalls for an Enterprise Deployment

Configuring Virtual Hosts on the Hardware Load Balancer 6-1
Overview of the Hardware Load Balancer Configuration 6-1
Typical Procedure for Configuring the Hardware Load Balancer 6-2
Summary of the Virtual Servers Required for an Enterprise Deployment 6-2
Additional Instructions for admin.example.com 6-3
Configuring the Firewalls and Ports for an Enterprise Deployment 6-3

7 Preparing the File System for an Enterprise Deployment

Overview of Preparing the File System for an Enterprise Deployment 7-1
Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment 7-2
About the Recommended Directory Structure for an Enterprise Deployment 7-3
File System and Directory Variables Used in This Guide 7-6
About Creating and Mounting the Directories for an Enterprise Deployment 7-10
Summary of the Shared Storage Volumes in an Enterprise Deployment 7-11

8 Preparing the Host Computers for an Enterprise Deployment

Verifying the Minimum Hardware Requirements for Each Host 8-2
Verifying Linux Operating System Requirements 8-2
Setting Linux Kernel Parameters 8-2
Setting the Open File Limit and Number of Processes Settings on UNIX Systems 8-3
Viewing the Number of Currently Open Files 8-3
Setting the Operating System Open File and Processes Limits 8-4

Verifying IP Addresses and Host Names in DNS or hosts File 8-4
Configuring Operating System Users and Groups 8-4
Enabling Unicode Support 8-5
Setting the DNS Settings 8-5
Configuring Users and Groups 8-5
Configuring a Host to Use an NTP (time) Server 8-7
Configuring a Host to Use an NIS/YP Host 8-8
Mounting the Required Shared File Systems on Each Host 8-9
Enabling the Required Virtual IP Addresses on Each Host 8-11

9 Preparing the Database for an Enterprise Deployment

Overview of Preparing the Database for an Enterprise Deployment 9-1
About Database Requirements 9-2
Supported Database Versions 9-2
Additional Database Software Requirements 9-2
Creating Database Services 9-3
Using SecureFiles for Large Objects (LOBs) in an Oracle Database 9-5
About Database Backup Strategies 9-6

Part III Configuring the Enterprise Deployment

10 Creating the Initial Oracle BI Domain for an Enterprise Deployment

Variables Used When Creating the BI Domain 10-2
Understanding the Initial BI Domain 10-3
About the Infrastructure Distribution 10-3
Characteristics of the Initial BI Domain 10-3
Installing the Oracle Fusion Middleware Infrastructure in Preparation for an Enterprise Deployment 10-4
Installing a Supported JDK 10-4
Locating and Downloading the JDK Software 10-4
Installing the JDK Software 10-4
Starting the Infrastructure Installer on BIHOST1 10-5
Navigating the Infrastructure Installation Screens 10-6
Checking the Directory Structure 10-7
Installing Oracle Business Intelligence in Preparation for an Enterprise Deployment 10-8
Starting the Installation Program 10-8
Navigating the Installation Screens 10-8
Checking the Directory Structure 10-10
Creating the Database Schemas 10-10

Installing and Configuring a Certified Database 10-11
Starting the Repository Creation Utility (RCU) 10-11
Navigating the RCU Screens to Create the Schemas 10-12
Configuring the BI Domain 10-14
Starting the Configuration Wizard 10-14
Navigating the Configuration Wizard Screens to Configure the BI Domain 10-14
Disabling the Derby Database 10-29
Creating the System Components on BIHOST1 10-29
Creating a BI Service Instance 10-30
Configuring the Singleton Data Directory (SDD) 10-32
Configuring Security for Essbase in Oracle Business Intelligence 10-33
Configuring the Domain Directories and Starting the Servers on BIHOST1 10-34
Starting the Node Manager in the Administration Server Domain Home on BIHOST1 10-35
Creating the boot.properties File 10-35
Starting the Administration Server Using the Node Manager 10-36
Validating the Administration Server 10-36
Creating a Separate Domain Directory for Managed Servers on BIHOST1 10-37
Starting the Node Manager in the Managed Server Domain Directory on BIHOST1 10-39
Starting the WLS_BI1 Managed Server on BIHOST1 10-40
Starting the System Components 10-41
Setting Up the Global Cache 10-42
Verifying Oracle Business Intelligence URLs on BIHOST1 10-44
Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group 10-44
About the Supported Authentication Providers 10-45
About the Enterprise Deployment Users and Groups 10-45
About Using Unique Administration Users for Each Domain 10-46
About the Domain Connector User 10-46
About Adding Users to the Central LDAP Directory 10-46
About Product-Specific Roles and Groups for Oracle Business Intelligence 10-47
Example Users and Groups Used in This Guide 10-47
Prerequisites for Creating a New Authentication Provider and Provisioning Users and Groups 10-48
Provisioning a Domain Connector User in the LDAP Directory 10-48
Creating the New Authentication Provider 10-49
Provisioning an Enterprise Deployment Administration User and Group 10-53
Adding the Administration Role to the New Administration Group 10-55
Adding weblogic_bi User to the BIServiceAdministrator Role 10-56
Updating the boot.properties File and Restarting the System 10-56
Adding the wsm-pm Role to the Administrators Group 10-56

Backing Up the Oracle Business Intelligence Configuration 10-57

11 Configuring Oracle HTTP Server for an Enterprise Deployment

Variables Used When Configuring the Oracle HTTP Server 11-2
About the Oracle HTTP Server Domains 11-2
Installing a Supported JDK 11-2
- Locating and Downloading the JDK Software 11-3
- Installing the JDK Software 11-3

Installing Oracle HTTP Server on WEBHOST1 11-4
- Starting the Installer on WEBHOST1 11-4
- Navigating the Oracle HTTP Server Installation Screens 11-4
- Verifying the Oracle HTTP Server Installation 11-6

Creating an Oracle HTTP Server Domain on WEBHOST1 11-7
- Starting the Configuration Wizard on WEBHOST1 11-7
- Navigating the Configuration Wizard Screens for an Oracle HTTP Server Domain 11-7

Installing and Configuring an Oracle HTTP Server Domain on WEBHOST2 11-9
Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2 11-10
- Starting the Node Manager on WEBHOST1 and WEBHOST2 11-10
- Starting the Oracle HTTP Server Instances 11-10

Configuring Oracle HTTP Server to Route Requests to the Application Tier 11-11
- About the Oracle HTTP Server Configuration for an Enterprise Deployment 11-11
  - Purpose of the Oracle HTTP Server Virtual Hosts 11-12
  - Recommended Structure of the Oracle HTTP Server Configuration Files 11-12
- Modifying the httpd.conf File to Include Virtual Host Configuration Files 11-12
- Creating the Virtual Host Configuration Files for Oracle Business Intelligence 11-13
- Configuring the WebLogic Proxy Plug-In 11-19
- Validating the Virtual Server Configuration on the Load Balancer 11-20
- Validating Access to the Management Consoles and Administration Server 11-20
- Validating HTTP Access to the Business Intelligence Components 11-20

Backing Up the Configuration 11-21

12 Scaling Out Oracle Business Intelligence

Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers 12-2
Installing Oracle Business Intelligence on the Other Host Computers 12-2
Stopping the Components on BIHOST1 12-2
- Stopping the System Components 12-3
- Stopping the WLS_BI1 Managed Server 12-4
- Stopping the Administration Server 12-5

Stopping the Node Manager in the Administration Server Domain Home 12-6
Stopping the Node Manager in the Managed Server Domain Directory 12-6
Cloning the Components on BIHOST1 12-6
Packing Up the Initial Domain on BIHOST1 12-7
Unpacking the Domain on BIHOST2 12-8
Starting the Components on BIHOST1 and BIHOST2 After Scaling Out 12-9
Starting the Node Manager in the Administration Server Domain Home 12-9
Starting the Administration Server 12-9
Starting the Node Managers in the Managed Server Domain Directories 12-10
Starting the Managed Servers 12-10
Starting the System Components 12-10
Verifying Oracle Business Intelligence URLs on BIHOST2 12-10
Configuring Oracle BI Publisher 12-11
Updating the JMS Shared Temp Directory 12-11
Configuring Integration with Oracle BI Presentation Services 12-12
Setting the Oracle BI EE Data Source 12-12
Configuring BIPJmsResource for the Oracle BI Cluster 12-14
Backing Up the Oracle Business Intelligence Configuration After Scaling Out 12-14

Part IV Common Configuration and Management Procedures for an Enterprise Deployment

13 Common Configuration and Management Tasks for an Enterprise Deployment

Verifying Manual Failover of the Administration Server 13-1
Failing Over the Administration Server to a Different Host 13-2
Validating Access to the Administration Server on BIHOST2 Through Oracle HTTP Server 13-4
Configuring Roles for Administration of an Enterprise Deployment 13-4
Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group 13-5
Failing the Administration Server Back to BIHOST1 13-5
Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer 13-7
When is SSL Communication Between the Middle Tier and Load Balancer Necessary? 13-7
Generating Self-Signed Certificates Using the utils.CertGen Utility 13-7
Creating an Identity Keystore Using the utils.ImportPrivateKey Utility 13-9
Creating a Trust Keystore Using the Keytool Utility 13-11
Importing the Load Balancer Certificate into the Truststore 13-11
Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts 13-12

Configuring Node Manager to Use the Custom Keystores 13-13
Configuring WebLogic Servers to Use the Custom Keystores 13-14
Performing Backups and Recoveries for an Enterprise Deployment 13-15
Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment 13-16
Products and Components that use JMS Persistence Stores and TLOGs 13-17
JDBC Persistent Stores vs. File Persistent Stores 13-17
About JDBC Persistent Stores for JMS and TLOGs 13-18
Performance Considerations for TLOGs and JMS Persistent Stores 13-18
Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment 13-20
Recommendations for TLOGs and JMS Datasource Consolidation 13-20
Roadmap for Configuring a JDBC Persistent Store for TLOGs 13-21
Roadmap for Configuring a JDBC Persistent Store for JMS 13-21
Creating a User and Tablespace for TLOGs 13-22
Creating a User and Tablespace for JMS 13-22
Creating GridLink Data Sources for TLOGs and JMS Stores 13-23
Assigning the TLOGs JDBC Store to the Managed Servers 13-25
Creating a JDBC JMS Store 13-26
Assigning the JMS JDBC store to the JMS Servers 13-26
Creating the Required Tables for the JMS JDBC Store 13-27
Using File Persistent Stores for TLOGs and JMS in an Enterprise Deployment 13-28
Configuring TLOGs File Persistent Store in a Shared Folder 13-28
Configuring JMS File Persistent Store in a Shared Folder 13-29
About JDBC Persistent Stores for Web Services 13-30
Performing Backups and Recoveries for an Enterprise Deployment 13-30
Setting the Front End Host and Port for a WebLogic Cluster 13-31

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

About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment 14-1
Understanding the Difference between Whole Server and Service Migration 14-1
Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment 14-2
Understanding Which Products and Components Require Whole Server Migration and Service Migration 14-3
Converting to Virtual IP Addresses in Preparation for Whole Server Migration 14-3
Enabling the Required Virtual IP Addresses on Each Host 14-3
Editing the Listen Address for the Managed Servers 14-3
Editing the OHS Virtual Host Configuration Files 14-4
Creating a GridLink Data Source for Leasing 14-4

Configuring Whole Server Migration for an Enterprise Deployment 14-7
Editing the Node Manager's Properties File to Enable Whole Server Migration 14-7
Setting Environment and Superuser Privileges for the wlsifconfig.sh Script 14-8
Setting the PATH Environment Variable for the wlsifconfig.sh Script 14-8
Granting Privileges to the wlsifconfig.sh Script 14-9
Configuring Server Migration Targets 14-9
Testing Whole Server Migration 14-10

15 Configuring Single Sign-On for an Enterprise Deployment

About Oracle HTTP Server Webgate 15-1
General Prerequisites for Configuring Oracle HTTP Server WebGate 15-2
Enterprise Deployment Prerequisites for Configuring OHS 12c Webgate 15-2
Configuring Oracle HTTP Server 12c WebGate for an Enterprise Deployment 15-3
Registering the Oracle HTTP Server WebGate with Oracle Access Manager 15-4
About RREG In-Band and Out-of-Band Mode 15-4
Updating the Standard Properties in the OAM11gRequest.xml File 15-5
Updating the Protected, Public, and Excluded Resources for an Enterprise Deployment 15-9
Running the RREG Tool 15-10
Running the RREG Tool in In-Band Mode 15-10
Running the RREG Tool in Out-Of-Band Mode 15-11
Files and Artifacts Generated by RREG 15-12
Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location 15-13
Insert OHS SimpleCA Certificate into the Wallet Artifact 15-14
Enable MD5 Certificate Signatures for the Oracle HTTP Server Instances 15-15
Restarting the Oracle HTTP Server Instance 15-16
Setting Up the WebLogic Server Authentication Providers 15-16
Backing Up Configuration Files 15-16
Setting Up the Oracle Access Manager Identity Assertion Provider 15-17
Updating the Default Authenticator and Setting the Order of Providers 15-17
Configuring Oracle ADF and OPSS Security with Oracle Access Manager 15-18
Configuring Single Sign-On for Applications 15-20
Enabling Single Sign-On and Oracle Access Manager for Oracle BI EE 15-20
Enabling Single Sign-On and Oracle Access Manager for Oracle BI Publisher 15-21

A Using Multi Data Sources with Oracle RAC

About Multi Data Sources and Oracle RAC A-1
Typical Procedure for Configuring Multi Data Sources for an Enterprise Deployment A-1

Preface

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

- Audience
- Documentation Accessibility
- 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 specialized administrators, such as database administrators (DBAs) and network administrators, where applicable.

Documentation Accessibility

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

Accessible Access to Oracle Support

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

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

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

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

- Enterprise Deployment Overview
  The Enterprise Deployment Guide provides detailed, validated instructions that help you plan, prepare, install, and configure a multi-host, secure, highly available, production topology for selected Oracle Fusion Middleware products.
- About a Typical Enterprise Deployment
  It is essential to understand the components of a typical enterprise deployment topology.
- Understanding the Business Intelligence Enterprise Deployment Topology
  The following topics describe the Oracle Business Intelligence enterprise deployment topologies.

1

Enterprise Deployment Overview

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

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

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

About the Enterprise Deployment Guide

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

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

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

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

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

When to Use the Enterprise Deployment Guide

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

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

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

2

About a Typical Enterprise Deployment

It is essential to understand the components of a typical enterprise deployment topology.

This chapter provides information on the Enterprise Deployment Topology diagram.

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

Diagram of a Typical Enterprise Deployment

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

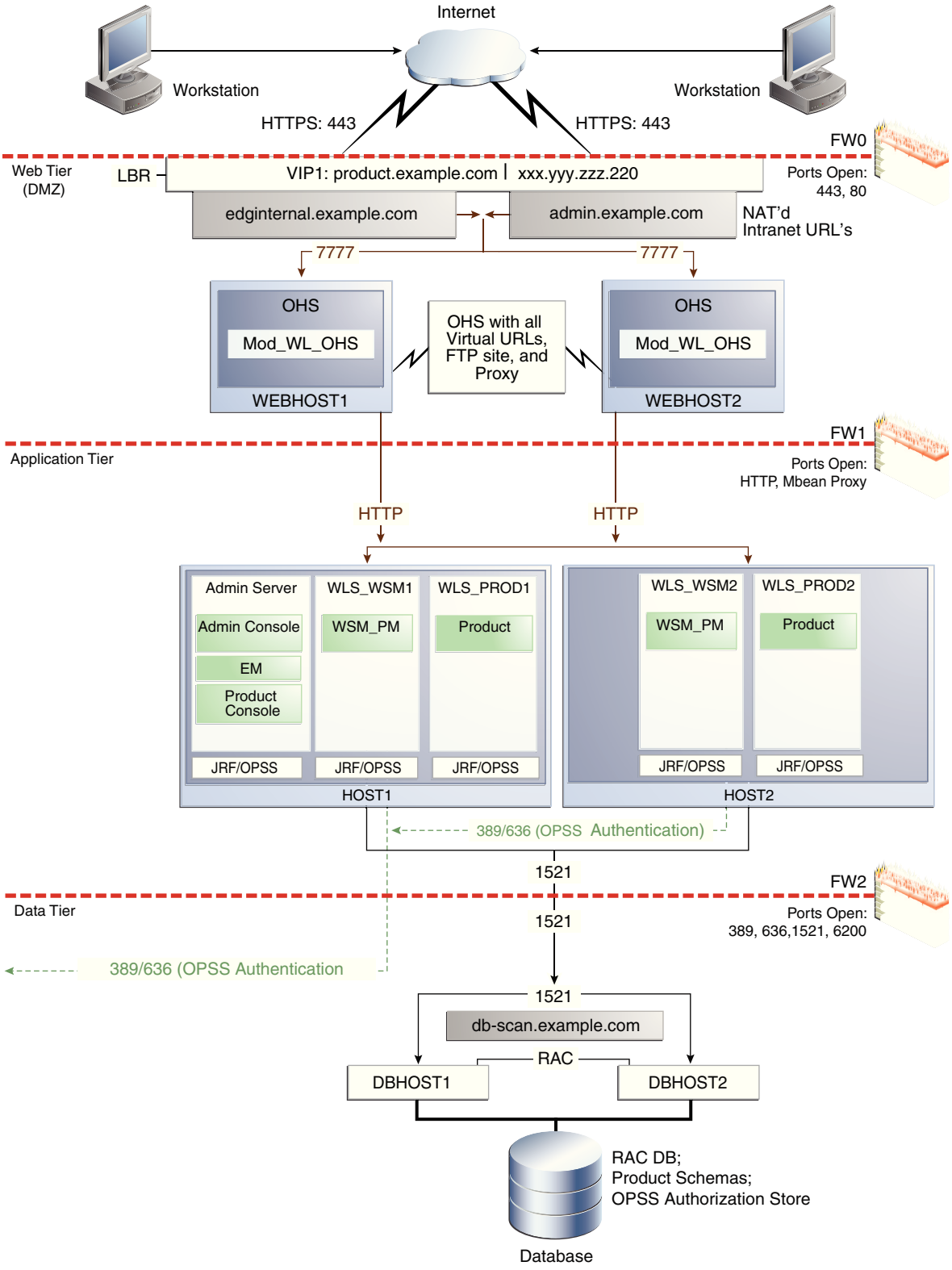
All Oracle Fusion Middleware enterprise deployments are designed to demonstrate the best practices for installing and configuring an Oracle Fusion Middleware production environment.

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

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

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

Figure 2-1 Typical Enterprise Deployment Topology Diagram



About the Typical Enterprise Deployment Topology Diagram

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

- Understanding the Firewalls and Zones of a Typical Enterprise Deployment
- 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

Understanding the Firewalls and Zones of a Typical Enterprise Deployment

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

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

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

For example:

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

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

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

  The ONS port (typically, 6200) is also required so that the application tier can receive notifications about workload and events in the Oracle RAC Database. These events are used by the Oracle WebLogic Server connection pools to adjust

quickly (creating or destroying connections), depending on the availability and workload on the Oracle RAC database instances.

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

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 perform internal invocations within the corporate network.
- A web tier, consisting of a hardware load balancer and two or more physical computers that host the web server instances (for high availability).

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

  The web server instances also host static web content that does not require the application logic to be delivered. Placing such content in the web tier reduces the overhead on the application servers and eliminates unnecessary network activity.
- An application tier, consisting of two or more physical computers that are hosting a cluster of Oracle WebLogic 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.

Receiving Requests Through Hardware Load Balancer

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

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

Purpose of the Hardware Load Balancer (LBR)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

For example:

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

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

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

- If the topology is active/passive DR:

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

- If the topology is active/active MDC:

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

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

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

Summary of the Typical Load Balancer Virtual Server Names

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

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

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

- `productinternal.example.com`: This virtual server name is for internal communications only.

  The load balancer uses its **Network Address Translation (NAT)** capabilities to route any internal communication from the application tier components that are directed to this URL. This URL is not exposed to external customers or users on the Internet. Each product has specific uses for the internal URL, so in the deployment instructions, the virtual server name is prefixed with the product name.

- `admin.example.com`: This virtual server name is for administrators who need to access the Oracle Enterprise Manager Fusion Middleware Control and Oracle WebLogic Server Administration Console interfaces.

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

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

HTTPS Versus HTTP Requests to the External Virtual Server Name

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

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

Understanding the Web Tier

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

- Benefits of Using a Web Tier to Route Requests

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

Benefits of Using a Web Tier to Route Requests

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

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

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

For more information about Oracle Traffic Director, see Getting Started with Oracle Traffic Director in *Administering Oracle Traffic Director*.

Alternatives to Using a Web Tier

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

This approach provide the following advantages:

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

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

Configuration of Oracle HTTP Server in the Web Tier

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

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

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

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

About Mod_WL_OHS

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

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

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:

- 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

Configuration of the Administration Server and Managed Servers Domain Directories

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

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

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

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

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

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

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

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

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

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

Using Oracle Web Services Manager in the Application Tier

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

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

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

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

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

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

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

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

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

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

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

About the Node Manager Configuration in a Typical Enterprise Deployment

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

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

About Using a Per Domain Node Manager Configuration

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

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

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

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

About Using a Per Host Node Manager Configuration

In a per host Node Manager configuration, you start a single Node Manager instance to control the Administration Server and all Managed Servers on a host, even those

that reside in different domains. This reduces the footprint and resource utilization on the Administration Server host, especially in those cases where multiple domains coexist on the same computer.

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

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

About Using Unicast for Communications within the Application Tier

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

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

Features of unicast in an enterprise deployment:

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

Features of multicast in an enterprise deployment:

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

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

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

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

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

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 Security Model

About the Data Tier

In the data tier, an Oracle RAC database runs on the two hosts (DBHOST1 and DBHOST2). The database contains the schemas required by the Oracle 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: See Introduction to Oracle Data Guard in *Oracle Data Guard Concepts and Administration*.
- Oracle RAC One Node: See Overview of Oracle RAC One Node in *Oracle Real Application Clusters Administration and Deployment Guide*.

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

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

3

Understanding the Business Intelligence Enterprise Deployment Topology

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

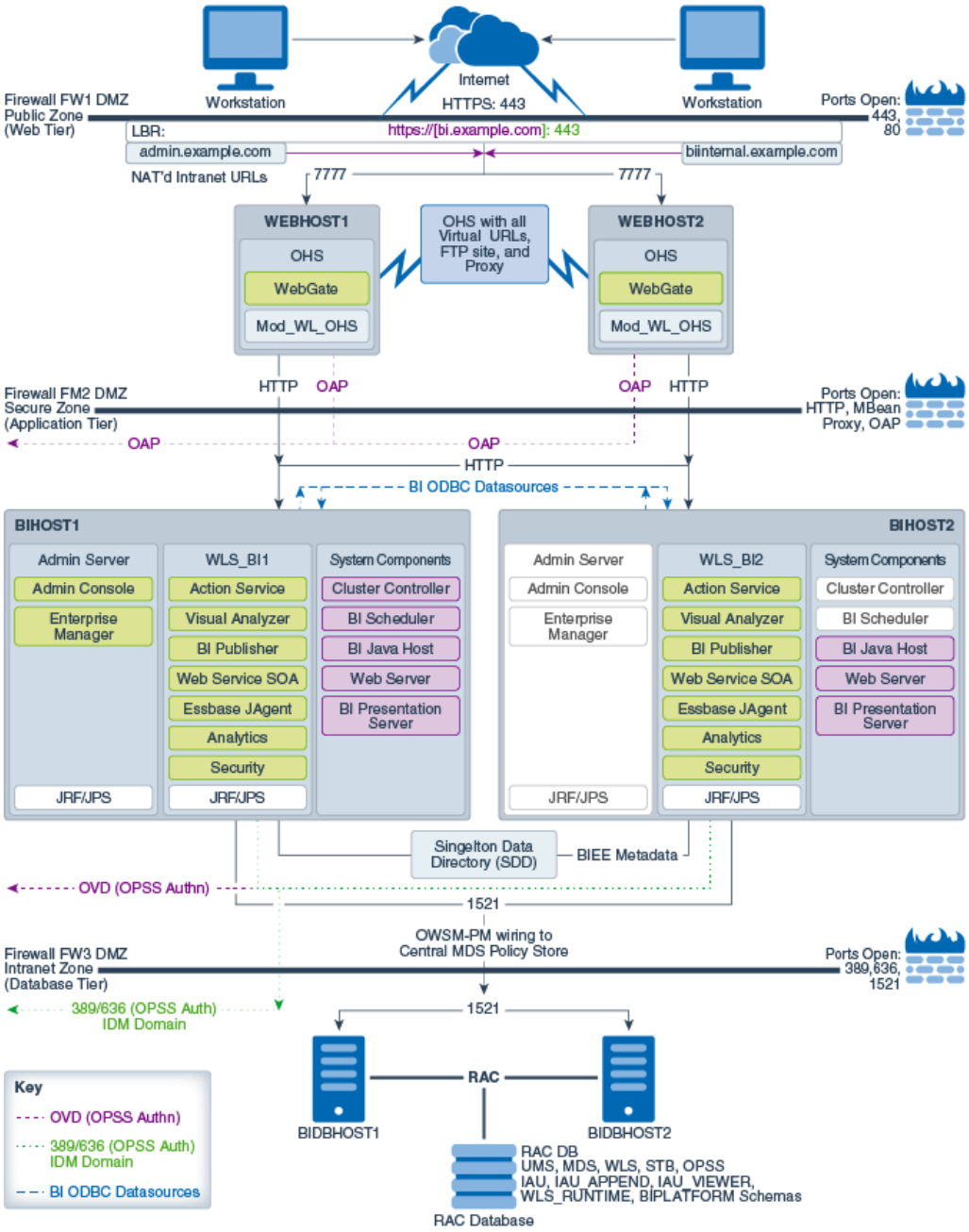
These topologies represent specific reference implementations of the concepts described in About a Typical Enterprise Deployment.

- Diagram of the Primary Business Intelligence Enterprise Topology
  This diagram shows the primary Oracle Business Intelligence enterprise deployment topology.
- Understanding the Primary Business Intelligence Topology Diagram
  This section provides information about the elements that are unique to the primary topology.
- 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.

Diagram of the Primary Business Intelligence Enterprise Topology

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

Figure 3-1 Primary Business Intelligence Enterprise Topology



Understanding the Primary Business Intelligence Topology Diagram

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 About a Typical Enterprise Deployment.

Before you review the information here, it is assumed you have reviewed the information in About 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.

- Summary of Business Intelligence Load Balancer Virtual Server Names
- Summary of the Managed Servers and Cluster on the Business Intelligence Application Tier

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.

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 activeactive 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

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.

- 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

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.

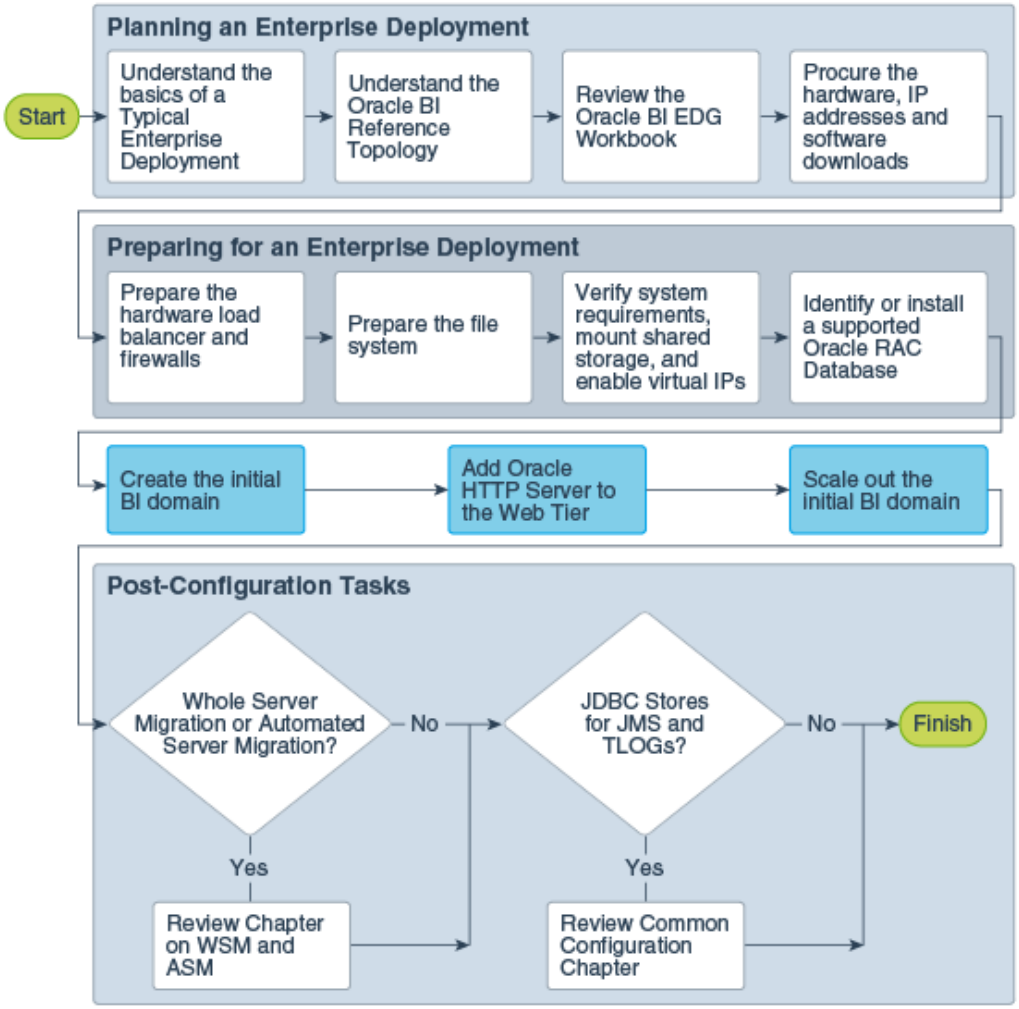
This guide is designed so you can start with a working Business Intelligence domain and then later scale out the domain to add additional capabilities.

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 3-2 Flow Chart of the Enterprise Topology Configuration Steps



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 that 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.
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

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 Oracle BI Domain for an Enterprise Deployment

Table 3-2 (Cont.) Roadmap Table for Configuring the Business Intelligence Enterprise Topology

Flow Chart Step	More Information
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

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

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

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

4

Using the Enterprise Deployment Workbook

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

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

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

Introduction to the Enterprise Deployment Workbook

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

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

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

Typical Use Case for Using the Workbook

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

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

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

Using the Oracle Business Intelligence Enterprise Deployment Workbook

Locating and understanding the Oracle Business Intelligence Enterprise Deployment workbook enables you to use it efficiently.

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

- Locating the Oracle Business Intelligence Enterprise Deployment Workbook
- Understanding the Contents of the Oracle Business Intelligence Enterprise Deployment Workbook

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.

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 that you need to install and configure the Enterprise Deployment topologies.

- Using the Start Tab
- Using the Hardware - Host Computers Tab
- Using the Network - Virtual Hosts & Ports Tab

- Using the Storage - Directory Variables Tab
- Using the Database - Connection Details Tab

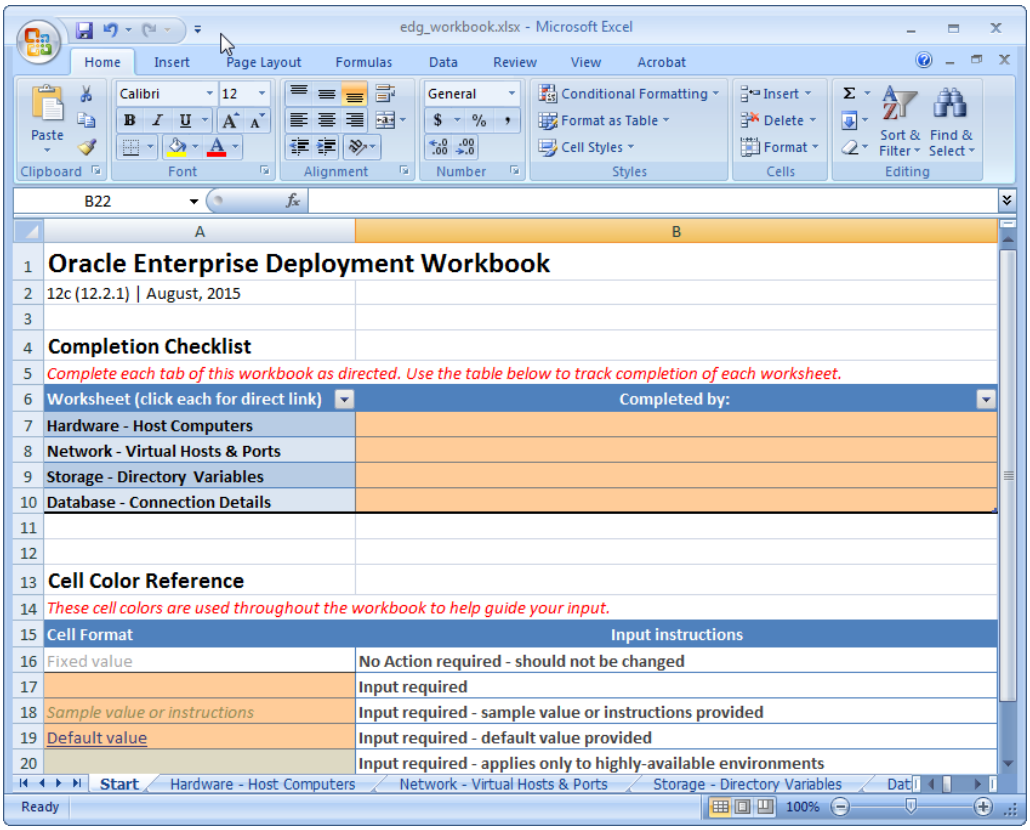
Using the Start Tab

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

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

The following image shows the Start tab of the Enterprise Deployment workbook.

Figure 4-1 Start Tab of the Enterprise Deployment workbook



Using the Hardware - Host Computers Tab

The Hardware - Host Computers tab lists the host computers that are 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 BIHOST1 variable with the actual name provided on the Hardware - Host Computers tab of the workbook.

> **Note:**
>
> If two domains share the same node, for example, if you set up the Oracle SOA suite, and then create MFT with its own domain, you have two domains on the same node. In this case, you use BIHOST1 and MFTHOST1 at the same time, one for each domain.

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. See Preparing the Host Computers for an Enterprise Deployment.

Using the Network - Virtual Hosts & Ports Tab

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

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

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

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

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

Using the Storage - Directory Variables Tab

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

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

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

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

Using the Database - Connection Details Tab

When you install and configure the enterprise deployment topology, you 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 need to provide to connect to the database from tools, such as the Configuration Wizard and the Repository Creation Utility.

To be sure that you have these values handy, use this tab to enter the actual values for these variables in your database installation. See Preparing the Database for an Enterprise Deployment.

Who Should Use the Enterprise Deployment Workbook?

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

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

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

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

5

Procuring Resources for an Enterprise Deployment

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

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

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

Hardware and Software Requirements for the Enterprise Deployment Topology

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

This section includes the following sections.

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

Hardware Load Balancer Requirements

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

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

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

Host Computer Hardware Requirements

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

It includes the following topics.

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

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

General Considerations for Enterprise Deployment Host Computers

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

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

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

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

Reviewing the Oracle Fusion Middleware System Requirements

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

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

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

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

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

This section specifies the typical memory, number of file descriptors, and operating system processes and tasks details 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 the minimum 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. 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
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	For the latest requirements for system components for the Oracle Fusion Middleware 12*c* (12.2.1.3.0) products, review the *Oracle Fusion Middleware System Requirements and Specifications*.		
TOTAL	11.0 GB*	17000	1200

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

Typical Disk Space Requirements for an Enterprise Deployment

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

For the latest disk space requirements for the Oracle Fusion Middleware 12*c* (12.2.1.3.0) 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 that is 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 n = number of disks, at least 4 (striped as one disk) 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.

Operating System Requirements for an Enterprise Deployment Topology

This section provides details about the operating system requirements.

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

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

Reserving the Required IP Addresses for an Enterprise Deployment

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

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

- Physical IP (IP) addresses for each of the host computers that you have procured for the topology

- A virtual IP (VIP) address for the Administration Server
- Additional VIP addresses for each Managed Server that is configured for Whole Server Migration

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

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

For more information, see the following topics.

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

What is a Virtual IP (VIP) Address?

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

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

Why Use Virtual Host Names and Virtual IP Addresses?

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

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

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

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

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

Physical and Virtual IP Addresses Required by the Enterprise Topology

This section describes the physical IP (IP) and virtual IP (VIP) addresses that are required for the Administration Server and each of the Managed Servers in a typical Oracle 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, each VIP is referenced by using the suggested host names in the table.

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

Table 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.
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. It is enabled in the node where the WLS_BI1 process is running (BIHOST1). Since BI only supports Whole Server Migration, this VIP is required if you are configuring BI for high availability.

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

Virtual IP	VIP Maps to...	Description
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. It is enabled in the node where the WLS_BI2 process is running (BIHOST2). Since BI only supports Whole Server Migration, this VIP is required if you are configuring BI for high availability.

Identifying and Obtaining Software Distributions for an Enterprise Deployment

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

The following table lists the distributions used in this guide.

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

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

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

Distribution	Description
Oracle Fusion Middleware Infrastructure 12*c* (12.2.1.3.0)	Download this distribution to install the Oracle Fusion Middleware Infrastructure, which includes Oracle WebLogic Server and Java Required Files software required for Oracle Fusion Middleware products. This distribution also installs the Repository Creation Utility (RCU), which in previous Oracle Fusion Middleware releases was packaged in its own distribution.

Distribution	Description
Oracle HTTP Server 12*c* (12.2.1.3.0)	Download this distribution to install the Oracle HTTP Server software on the Web Tier.
Oracle Business Intelligence 12*c* (12.2.1.3.0)	Download this distribution to install the Oracle Business Intelligence software.

6

Preparing the Load Balancer and Firewalls for an Enterprise Deployment

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

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

Configuring Virtual Hosts on the Hardware Load Balancer

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

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

- Overview of the Hardware Load Balancer Configuration
- Typical Procedure for Configuring the Hardware Load Balancer
- Summary of the Virtual Servers Required for an Enterprise Deployment
- Additional Instructions for admin.example.com

Overview of the Hardware Load Balancer Configuration

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

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

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

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

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

Typical Procedure for Configuring the Hardware Load Balancer

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

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

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

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

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

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

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

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

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

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

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

Summary of the Virtual Servers Required for an Enterprise Deployment

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

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

Table 6-1 Virtual Servers Required for an Enterprise Deployment

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

Additional Instructions for admin.example.com

This section provides additional instructions that are 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.

Configuring the Firewalls and Ports for an Enterprise Deployment

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

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

Firewall notation:

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

Table 6-2 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.

Table 6-2 (Cont.) Firewall Ports Common to All Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Browser request	FW0	443	HTTPS / Load Balancer	Inbound	Timeout depends on the size and type of HTML content.
Browser request	FW1	80	HTTP / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Browser request	FW1	443	HTTPS / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	80	HTTP / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	443	HTTPS / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Load balancer to Oracle HTTP Server	n/a	7777	HTTP	n/a	n/a
OHS registration with Administration Server	FW1	7001	HTTP/t3	Inbound	Set the timeout to a short period (5-10 seconds).
OHS management by Administration Server	FW1	OHS Admin Port (7779)	TCP 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.

Table 6-2 (Cont.) Firewall Ports Common to All Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
Administration Console access	FW1	7001	HTTP / Administration Server and Enterprise Manager t3	Both	You should tune this timeout based on the type of access to the admin console (whether 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 Range: 8000 - 8090		n/a	n/a
Oracle Unified Directory access	FW2	389 636 (SSL)	LDAP or LDAP/ssl	Inbound	You should tune the directory server's parameters based on load balancer, and not the other way around.
Oracle Notification Server (ONS)	FW2	6200	ONS	Both	Required for Gridlink. An ONS server runs on each database server.

Table 6-3 Firewall Ports for Product-specific Components in Oracle Fusion Middleware Enterprise Deployments

Type	Firewall	Port and Port Range	Protocol / Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
WSM-PM access	FW1	7010 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

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

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

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

Overview of Preparing the File System for an Enterprise Deployment

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

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

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

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

Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment

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

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

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

Table 7-1 lists the key sections that 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. For an enterprise deployment, Oracle recommends that you use redundant Oracle homes on separate storage volumes. If a separate volume is not available, a separate partition on the shared disk should be used to provide redundant Oracle homes to application tier hosts.
Using Shared Storage for Domain Configuration Files	Describes the concept of creating separate domain homes for the Administration Server and the Managed Servers in the domain. For an enterprise deployment, the Administration Server domain home location is referenced by the *ASERVER_HOME* variable.
Shared Storage Requirements for JMS Stores and JTA Logs	Provides instructions for setting the location of the transaction logs and JMS stores for an enterprise deployment.
Introduction to Zero Downtime Patching	Describes the Zero Downtime feature and the procedure to configure and monitor workflows.

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

> **Note:**
>
> Oracle Business Intelligence has an additional shared storage requirement for setting the location of specific Business Intelligence metadata. See Configuring the Singleton Data Directory (SDD).

About the Recommended Directory Structure for an Enterprise Deployment

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

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

The diagrams are used to indicate:

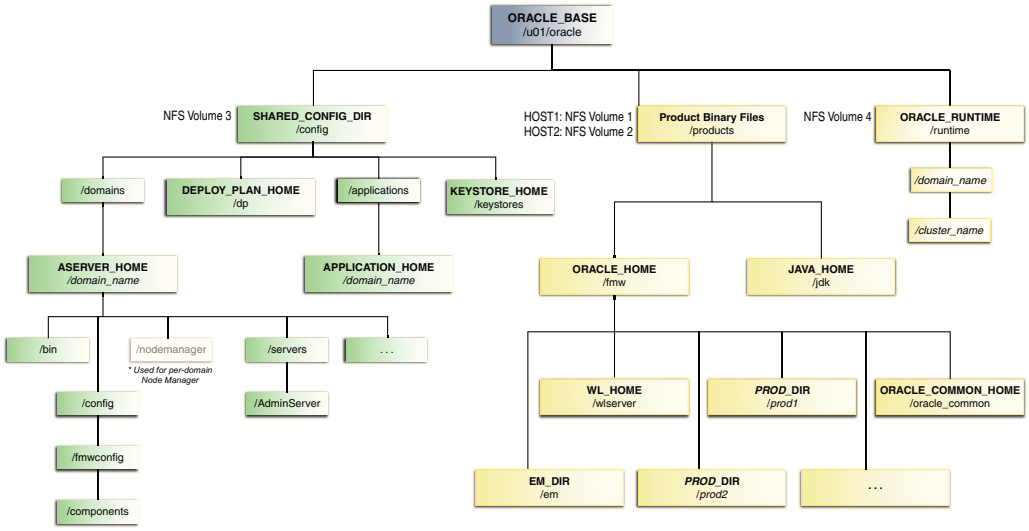
- Figure 7-1 shows the resulting directory structure on the shared storage device after you have installed and configured a typical Oracle Fusion Middleware enterprise deployment. The shared storage directories are accessible by the application tier host computers.
- Figure 7-2 shows the resulting directory 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.

> **Note:**
>
> Figure 7-3 assumes that you are using Oracle HTTP Server in the web tier. However, you can also use Oracle Traffic Director to route HTTP and other requests to the application tier.

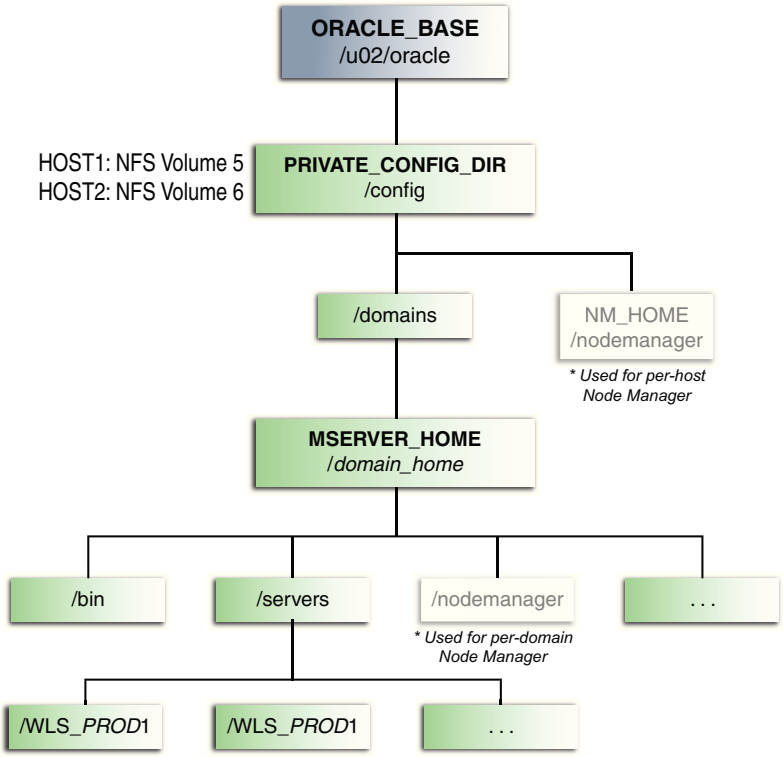
Where applicable, the diagrams also include the standard variables used to reference the directory locations in the installation and configuration procedures in this guide.

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



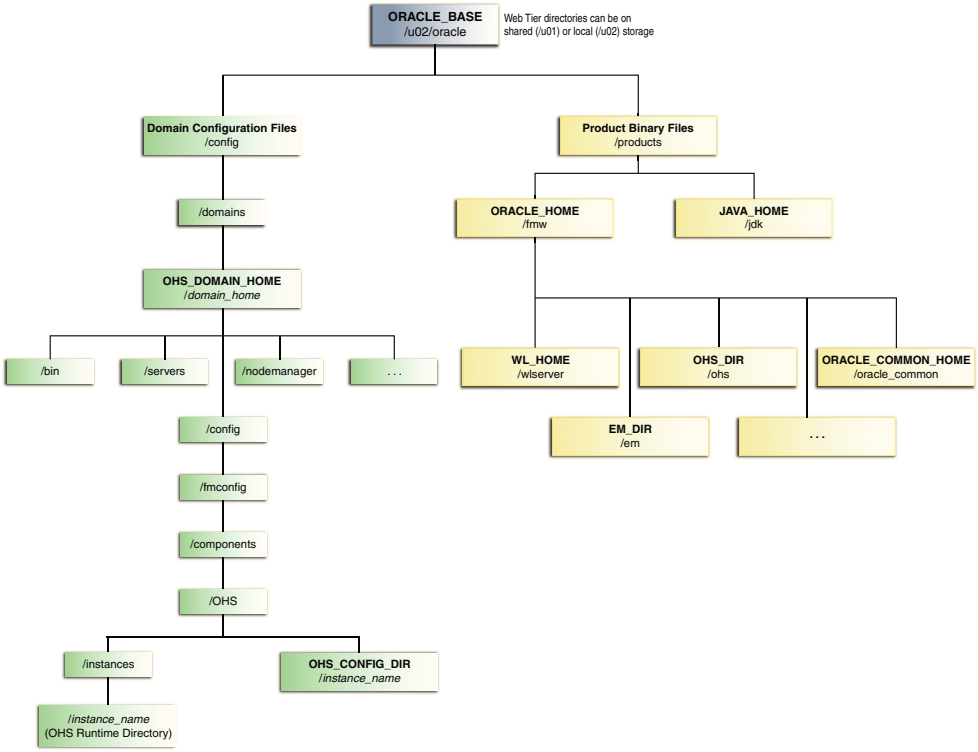
*See About the Node Manager Configuration in a Typical Enterprise Deployment.

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



* 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



File System and Directory Variables Used in This Guide

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

Table 7-2 lists the file system directories and the directory variables that are used to reference the directories on the application tier. Table 7-3 lists the file system directories and variables that are used to reference the directories on the web tier.

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

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

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

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

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

Directory Variable	Description	Relative Path	Sample Value on the Application Tier
ORACLE_BASE	The base directory, under which Oracle products are installed.	N/A	`/u01/oracle`
ORACLE_HOME	The read-only location for the product binaries. For the application tier host computers, it is stored on shared disk. The Oracle home is created when you install the Oracle Fusion Middleware Infrastructure software. You can then install additional Oracle Fusion Middleware products into the same Oracle home.	*ORACLE_BASE*`/products/fmw`	`/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.	*ORACLE_HOME*`/oracle_common`	`/u01/oracle/products/fmw/oracle_common`
WL_HOME	The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored.	*ORACLE_HOME*`/wlserver`	`/u01/oracle/products/fmw/wlserver`
PROD_DIR	Individual product directories for each Oracle Fusion Middleware product that you install.	*ORACLE_HOME*`/prod_dir`	`/u01/oracle/products/fmw/prod_dir` The product can be `soa`, `wcc`, `idm`, `bi`, or another value, depending on your enterprise deployment.
EM_DIR	The product directory used to store the Oracle Enterprise Manager Fusion Middleware Control software binaries.	*ORACLE_HOME*`/em`	`/u01/oracle/products/fmw/em`
JAVA_HOME	The location where you install the supported Java Development Kit (JDK).	*ORACLE_BASE*`/products/jdk`	`/u01/oracle/products/jdk`

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

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

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

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

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

Directory Variable	Description	Sample Value on the Web Tier
WEB_ORACLE_HOME	The read-only location for the Oracle HTTP Server product binaries. For the web tier host computers, this directory is stored on the local disk. The Oracle home is created when you install the Oracle HTTP Server software .	`/u02/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 that you install.	`/u02/oracle/products/fmw/ohs`
JAVA_HOME	The location where you install the supported Java Development Kit (JDK).	`/u02/oracle/products/jdk`
WEB_DOMAIN_HOME	The Domain home for the standalone Oracle HTTP Server domain, which is created when you install Oracle HTTP Server on the local disk of each web tier host.	`/u02/oracle/config/domains/`*`domain_name`* In this example, replace *`domain_name`* with the name of the WebLogic Server domain.

WEB_CONFIG_DIR	This is the location where you edit the Oracle HTTP Server configuration files (for example, `httpd.conf` and `moduleconf/*.conf`) on each web host.
	Note this directory is also referred to as the OHS Staging Directory. Changes made here are later propagated to the OHS Runtime Directory.
	See Staging and Run-time Configuration Directories in the *Administering Oracle HTTP Server*.
		`/u02/oracle/config/domains`
		`/`*`domain_name`*`/config/fmwconfig`
		`/components/OHS/`*`instance`*`/`
		`/`*`instance_name`*

WEB_KEYSTORE_HOME	If you use Oracle Traffic Director as your web server, this is the location for custom certificates and keystores.	`/u02/oracle/config/keystores`

About Creating and Mounting the Directories for an Enterprise Deployment

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

- For the application tier, install the Oracle home, which contains the software binaries, on a second shared storage volume or second partition that is mounted

to 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/
```

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

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

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

  As with the application tier servers (BIHOST1 and BIHOST2), use the same directory path on both computers.

  For example:

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

Summary of the Shared Storage Volumes in an Enterprise Deployment

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

You can use shared storage to host the Web tier binaries and config to make backups easier so that files are stored on a more fault-tolerant hardware, but each node needs to use a private directory that is not shared with the other nodes.

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

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

Table 7-4 Shared Storage Volumes in an Enterprise Deployment

Volume in Shared Storage	Mounted to Host	Mount Directories	Description and Purpose
NFS Volume 1	BIHOST1	`/u01/oracle/products/`	Storage for the product binaries to be used by BIHOST1; this is where the Oracle home directory and product directories are installed. Used initially by BIHOST1, but can be shared with other hosts when scaling-out the topology.

Table 7-4 (Cont.) Shared Storage Volumes in an Enterprise Deployment

Volume in Shared Storage	Mounted to Host	Mount Directories	Description and Purpose
NFS Volume 2	BIHOST2	`/u01/oracle/products/`	Storage for the product binaries to be used by BIHOST2; this is where the Oracle home directory and product directories are installed. Used initially by BIHOST2, but can be shared with other hosts when scaling-out the topology.
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 or the Oracle Traffic Director software binaries (Oracle home) and domain configuration files that are used by WEBHOST1, if the web tier private binary and config directories reside on shared storage.

Table 7-4 (Cont.) Shared Storage Volumes in an Enterprise Deployment

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 or the Oracle Traffic Director software binaries (Oracle home) and domain configuration files that are used by WEBHOST2, if the Web Tier private binary and config directories reside on shared storage.

8

Preparing the Host Computers for an Enterprise Deployment

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

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

- Verifying the Minimum Hardware Requirements for Each Host
  After you procure the required hardware for the enterprise deployment, it is important to ensure that each host computer meets the minimum system requirements.
- Verifying Linux Operating System Requirements
  You can review the typical Linux operating system settings for an enterprise deployment in this section.
- Configuring Operating System Users and Groups
  The users and groups to be defined on each of the computers that host the enterprise deployment are listed in this section.
- Enabling Unicode Support
  It is recommended to enable Unicode support in your operating system so as to allow processing of characters in Unicode.
- Setting the DNS Settings
- Configuring Users and Groups
- Configuring a Host to Use an NTP (time) Server
- Configuring a Host to Use an NIS/YP Host
- Mounting the Required Shared File Systems on Each Host
  It is important to understand how to mount the shared storage to all the servers that require access.
- Enabling the Required Virtual IP Addresses on Each Host
  You must enable the required virtual IP addresses on each host in order to prepare the host for the enterprise deployment.

Verifying the Minimum Hardware Requirements for Each Host

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

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

If you deploy 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 the `/tmp` directory.

Verifying Linux Operating System Requirements

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

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

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

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

Setting Linux Kernel Parameters

The kernel-parameter and shell-limit values shown in Table 8-1are 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 8-1 on all nodes in the topology.

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

If you deploy a database onto the host, you might need to modify additional kernel parameters. Refer to 12*c* (12.2.1.3.0) Configuring Kernel Parameters in *Oracle Grid Infrastructure Installation Guide for Linux*.

Table 8-1 UNIX Kernel Parameters

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

To set these parameters:

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

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

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

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

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

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

For more information, see the following sections.

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

Viewing the Number of Currently Open Files

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

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

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

C shell:

```
limit descriptors
```

Bash:

```
ulimit -n
```

Setting the Operating System Open File and Processes Limits

To change the Open File Limit values:

1. Sign in as `root` user 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.

> **Note:**
>
> If you are running Oracle Enterprise Linux 6 or Red Hat Linux 6, locate the following operating system configuration file:`/etc/security/limits.d/90-nproc.conf`
>
> Ensure that the same values are added to the `90-nproc.conf` file. Otherwise, the values in the `90-nproc.conf` file overrides the values in the `limits.conf` file.

4. Re-login into the host computer.

Verifying IP Addresses and Host Names in DNS or hosts File

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

```
IP_Address Fully_Qualified_Name Short_Name
```

For example:

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

Configuring Operating System Users and Groups

The users and groups to be defined on each of the computers that host the enterprise deployment are listed in this section.

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.

Enabling Unicode Support

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

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

On UNIX operating systems, Oracle highly recommends that you enable Unicode support by setting the LANG and LC_ALL environment variables to a locale with the UTF-8 character set. This enables the operating system to process any character in Unicode. Oracle 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.

Setting the DNS Settings

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

Configuring Users and Groups

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

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

Groups

You must create the following groups on each node.

- `oinstall`
- `dba`

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

```
groupadd groupname
```

For example

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

User

You must create the following user on each node.

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

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

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

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

For example:

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

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

Configuring a Host to Use an NTP (time) Server

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

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

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

   > **Note:**
   >
   > If desired, add additional peer entries to `/etc/ntp.conf` for each of the app-tier hosts, DB hosts, and storage appliance. This enables additional redundancy and accuracy in the event of loss or inaccuracy of the NTP servers. Add one entry per host as follows:
   >
   > ```
   > # # enable app-to-app host time sync redundancy
   > peer BIHOST1
   > peer BIHOST2
   > peer BIHOST1
   > peer BIHOST2
   > # enable app-to-db host time sync redundancy, if permitted
   > peer DBHOST1
   > peer DBHOST2
   > # enable app-to-Storage time sync redundancy, if permitted
   > peer NFSAPLIANCE
   > ```

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

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

5. Start the NTP client using the following command:

   ```
   service ntpd start
   ```

6. Validate that NTP time synchronization is active, the NTP servers and app-tier hosts are reachable, and the time offset (milliseconds) is not significantly high for any listed host. Run the `ntpq` command as follows:

   ```
   /usr/sbin/ntpq -p

   remote           refid      st t when poll reach   delay
   offset  jitter
   ==============================================================================
   +ntphost1.example.com 10.1.1.13   3  u    1   16  377    0.549
   0.032   0.041
   ```

```
*ntphost2.example.com 10.1.2.10   2  u     11   16    377    0.349
0.079   0.039
-wcchost1.example.com 10.1.2.200  3  u     54   64    376    0.124
-0.068  3.681
-wcchost2.example.com 10.1.2.201  3  u     66   64    376    0.059
-0.123  2.038
-wcphost1.example.com 10.1.2.205  3  u     18   64    377    0.050
-0.091  2.930
#wcphost2.example.com 10.1.2.206  3  u     25   64    376    0.041
-0.105  4.919
```

> **Note:**
>
> It may take several minutes for all servers to register and coordinate NTP sync once fully configured. Run the `ntpq` command periodically until assured that all servers and peers are synchronizing their time.

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

   ```
   chkconfig ntpd on
   ```

Configuring a Host to Use an NIS/YP Host

If you are using NFS Version 4, configure a directory service or an NIS (Network Information Server). If your organization does not have one already, use the built-in one on the ZFS storage appliance. See Configuring NFS Version 4 (NFSv4) on Exalogic in the *Oracle Fusion Middleware Exalogic Machine Owner's Guide* for more information.

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

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

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

   Set the domain value, as in the following example:

   ```
   Domain = example.com
   ```

3. Restart the `rpcidmapd` service:

   ```
   service rpcidmapd restart
   ```

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

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

   Add the following line:

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

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

5. Set NIS domain name on the command line:

   ```
   domainname NIS_DOMAIN_NAME
   ```

   For example:

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

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

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

   Add `nis` to each of the following entries:

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

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

7. Restart the `rpcidmapd` service:

   ```
   service rpcidmapd restart
   ```

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

   ```
   service ypbind restart
   ```

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

   ```
   ypwhich
   ```

10. Verify if you can access Oracle user accounts:

    ```
    ypcat passwd
    ```

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

    ```
    chkconfig ypbind on
    ```

Mounting the Required Shared File Systems on Each Host

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

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

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

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

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

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

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

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

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

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

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

To mount the shared storage on Linux:

1. Create the mount directories on 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 that you just configured.

For example:

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

Verify that the owner and permissions are correct:

```
$ ls -l testfile
```

Then remove the file:

```
$ rm testfile
```

> **Note:**
>
> The shared storage can be a NAS or SAN device. The following example illustrates creating storage for a NAS device from 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 to learn about the appropriate options for your environment.

Enabling the Required Virtual IP Addresses on Each Host

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

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

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

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

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

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

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

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

   ```
   ip addr show dev eth0
   ```

Sample output:

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

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

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

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

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

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

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

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

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

9

Preparing the Database for an Enterprise Deployment

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

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

- Overview of Preparing the Database for an Enterprise Deployment
  It is important to understand how to configure a supported database as part of an Oracle Fusion Middleware enterprise deployment.
- About Database Requirements
  Before you configure the enterprise deployment topology, you have to verify that the database meets the requirements described in the following sections.
- Creating Database Services
  When multiple Oracle Fusion Middleware products are sharing the same database, each product should be configured to connect to a separate, dedicated database service. This service should be different from the default database service. Having a different service name from the default, allows you to create role based database services for Disaster Recovery and Multi-Datacenter topologies.
- Using SecureFiles for Large Objects (LOBs) in an Oracle Database
  SecureFiles is a new LOB storage architecture introduced in Oracle Database 11g Release 1. It is recommended to use SecureFiles for the Oracle Fusion Middleware schemas, in particular for the Oracle SOA Suite schemas.
- About Database Backup Strategies
  Performing a database backup at key points in the installation and configuration of an enterprise deployment enables you to recover quickly from any issue that might occur in the later configuration steps.

Overview of Preparing the Database for an Enterprise Deployment

It is important to understand 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 by using the Oracle Fusion Middleware Repository Creation Utility (RCU).

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

About Database Requirements

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

- Supported Database Versions
- Additional Database Software Requirements

Supported Database Versions

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

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

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

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

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

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

Additional Database Software Requirements

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

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

- **Oracle Clusterware**

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

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

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

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

- **Automatic Storage Management** (optional)

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

Creating Database Services

When multiple Oracle Fusion Middleware products are sharing the same database, each product should be configured to connect to a separate, dedicated database service. This service should be different from the default database service. Having a different service name from the default, allows you to create role based database services for Disaster Recovery and Multi-Datacenter topologies.

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

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

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

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`
> - Enter the `EXECUTE DBMS_SERVICE` command shown on a single line.
>
>   For more information about the `DBMS_SERVICE` package, see DBMS_SERVICE in *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 -db bidb -service biedg.example.com -preferred
   bidb1,bidb2
   ```

3. Start the service:

   ```
   srvctl start service –db bidb –service 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.

   For example:

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

   ```
   srvctl config service -db bidb -service biedg.example.com
   ```

   Several parameters are shown. Check the following parameters:

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

   You can modify these parameters by using the following command:

   ```
   srvctl modify service -db bidb -service biedg.example.com -rlbgoal
   SERVICE_TIME -clbgoal SHORT
   ```

   Restart the service:

   ```
   srvctl stop service -db bidb -service biedg.example.com
   srvctl start service -db bidb -service biedg.example.com
   ```

Verify the change in the configuration:

```
srvctl config service -db bidb -service biedg.example.com
Runtime Load Balancing Goal: SERVICE_TIME
  Service name: biedg.example.com
  Service is enabled
  Server pool: bidb_biedg.example.com
  ...
  Connection Load Balancing Goal: SHORT
  Runtime Load Balancing Goal: SERVICE_TIME
  ...
```

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

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

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

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

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

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

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

Other values for the `db_securefile` parameter are:

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

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

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

About Database Backup Strategies

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

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

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

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

Part III
Configuring the Enterprise Deployment

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

- Creating the Initial Oracle 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.
- Configuring Oracle HTTP Server for an Enterprise Deployment
  When you configure the web tier, you have the option of using Oracle HTTP Server or Oracle Traffic Director. If you choose to use Oracle HTTP Server, then you must install Oracle HTTP Server on each of the web tier hosts and configure Oracle HTTP standalone domains on each host.
- Scaling Out Oracle Business Intelligence
  This chapter describes the steps to scale out your initial Oracle Business Intelligence domain to BIHOST2.

10

Creating the Initial Oracle 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 Oracle BI domain, creating database schemas and configuring the Oracle BI domain.

- 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.
- Understanding the Initial BI Domain
  Before you being creating the initial Business Intelligence (BI) domain, be sure to review the following key concepts.
- 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.
- 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.
- 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.
- Configuring the BI Domain
  This section provides instructions for creating a WebLogic domain using the configuration wizard.
- Disabling the Derby Database
- 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.
- Creating a BI Service Instance
  Perform the steps in this section to create a new BI Service instance.
- 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.
- 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.

- 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.
- 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.
- 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.

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
- DBHOST1
- DBHOST2
- BIHOST1

- SCAN Address for the Oracle RAC Database (DB-SCAN.example.com)
- BIHOST1VHN
- BIHOST2VHN

Understanding the Initial BI Domain

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

- About the Infrastructure Distribution
- Characteristics of the Initial BI Domain

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.

See About the Oracle Fusion Middleware Infrastructure in *Understanding Oracle Fusion Middleware*.

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

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.

- Installing a Supported JDK
- Starting the Infrastructure Installer on BIHOST1
- Navigating the Infrastructure Installation Screens
- Checking the Directory Structure
  After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.

Installing a Supported JDK

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

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

Locating and Downloading the JDK Software

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

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

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the following locations:

- On the shared storage device, install the JDK in the `/u01/oracle/products/jdk` directory. The JDK 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 Understanding the Recommended Directory Structure for an Enterprise Deployment.

To install JDK 1.8.0_131:

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

   ```
   cd download_dir
   ```

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

   ```
   tar -xzvf jdk-8u131-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_131 /u01/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

   For example:

   ```
   export JAVA_HOME=/u01/oracle/products/jdk
   export PATH=$JAVA_HOME/bin:$PATH
   ```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

   ```
   java -verison
   ```

   The Java version in the output should be displayed as “1.8.0_131”.

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 following example:

   ```
   $JAVA_HOME/bin/java -d64 -jar distribution_file_name.jar
   ```

   In this example:

   - Replace *JAVA_HOME* with the environment variable or actual JDK location on your system.

- Replace *distribution_file_name* with the actual name of the distribution JAR file.

  If you download the distribution from the Oracle Technology Network (OTN), then the JAR file is typically packaged inside a downloadable compressed file.

  To install the software required for the initial Infrastructure domain, the distribution you want to install is **fmw_12.2.1.3.0_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.

Navigating the Infrastructure Installation Screens

The installation program displays a series of screens, in the order listed in the following table.

If you need additional help with any of the installation screens, click the screen name or click the **Help** button on the screen.

Table 10-2 Navigating the Infrastructure Installation Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears if you are installing any Oracle product on this host for the first time. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location. See Understanding the Oracle Central Inventory in *Installing Software with the Oracle Universal Installer*. **Note:** Oracle recommends that you configure the central inventory directory on the products shared volume. Example: `/u01/oracle/products/oraInventory` You may also need to execute the `createCentralinventory.sh` script as root from the `oraInventory` folder after the installer completes.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to search My Oracle Support automatically for available patches or automatically search a local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For the purposes of an enterprise deployment, enter the value of the *ORACLE_HOME* variable listed in Table 7-2.

Table 10-2 (Cont.) Navigating the Infrastructure Installation Screens

Screen	Description
Installation Type	Use this screen to select the type of installation and as a consequence, the products and feature sets that you want to install. For this topology, select **Fusion Middleware Infrastructure**. **Note:** The topology in this document does not include server examples. Oracle strongly recommends that you do not install the examples into a production environment.
Prerequisite Checks	This screen verifies that your system meets the minimum requirements. If there are any warning or error messages, refer to the Oracle Fusion Middleware System Requirements and Specifications document on the Oracle Technology Network (OTN).
Security Updates	If you already have an Oracle Support account, use this screen to indicate how you would like to receive security updates. If you do not have one and are sure that you want to skip this step, clear the check box and verify your selection in the follow-up dialog box.
Installation Summary	Use this screen to verify the installation options that you have selected. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file. Response files can be used later in a silent installation situation. For more information about silent or command-line installation, see Using the Oracle Universal Installer in Silent Mode in *Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click **Finish** to dismiss the installer.

Checking the Directory Structure

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

To check the directory structure:

1. Change to the *ORACLE_HOME* directory where you installed the Infrastructure.
2. Enter the following command:

```
ls --format=single-column
```

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

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

See What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

Installing Oracle 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.

- Starting the Installation Program
- Navigating the Installation Screens
- Checking the Directory Structure
  After you install Oracle BI, you should see the directory structure as shown in this topic. The contents of your installation vary based on the options you selected during the installation.

Starting the Installation Program

Use these steps to start the BI Installer.

1. Sign in to BIHOST1.
2. Change to the directory where you downloaded the installation program.
3. Start the installation program by invoking the executable, as shown in the following example.

   (UNIX) `./ fmw_12.2.1.3.0_bi_platform_linux64.bin`

   (Windows) `fmw_12.2.1.3.0_bi_platform_win64.exe`

For more information about the actual file names for each distribution, see Identifying and Obtaining Software Distributions for an Enterprise Deployment.

When the installation program appears, you are ready to begin the installation. See Navigating the Installation Screens for a description of each installation program screen.

Navigating the Installation Screens

The installation program displays a series of screens, in the order listed in Table 10-3.

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

Table 10-3 Oracle Business Intelligence Install Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears 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. 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. 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. For this topology, select **BI Platform Distribution with Samples**.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, refer to the Oracle Fusion Middleware System Requirements and Specifications document on the Oracle Technology Network (OTN).
Auto Updates - Patch Selection	This screen appears if the both of the following statements are true: • You searched for available patches earlier in the installation session, using the Auto Updates screen. • 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. 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. 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 close the installer.

Checking the Directory Structure

After you install Oracle BI, you should see the directory structure as shown in this topic. The contents of your installation vary based on the options you selected during the installation.

To see the directory structure:

1. Change to the *ORACLE_HOME* directory where you installed Oracle BI.
2. Enter the following command:

   ```
   ls --format=single-column
   ```

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

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

   bi-epm-registry
   bifoundation
   bin
   clients
   common
   endpointmanager
   file_templates
   jlib
   lib
   modules
   nls
   oracore
   plugins
   products
   schema
   upgrade
   xsd
   ```

See What are the Key Oracle Fusion Middleware Directories? in *Understanding Oracle Fusion Middleware*.

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*.

- Installing and Configuring a Certified Database
- Starting the Repository Creation Utility (RCU)
- Navigating the RCU Screens to Create the Schemas

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*.

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Set the *JAVA_HOME* environment variable so it references the location where you installed a supported JDK.

   See File System and Directory Variables Used in This Guide.

2. Navigate to the following directory on BIHOST1:

   ```
   ORACLE_HOME/oracle_common/bin
   ```

3. Start RCU:

   ```
   ./rcu
   ```

> **Note:**
>
> If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces created by the RCU, provide the `-encryptTablespace true` option when you start the RCU.
>
> This will default the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See Encrypting Tablespaces in *Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

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

- Task 1, Introducing RCU
- Task 2, Selecting a Method of Schema Creation
- Task 3, Providing Database Credentials
- Task 4, Specifying a Custom Prefix and Selecting Schemas
- Task 5, Specifying Schema Passwords
- Task 6, Completing Schema Creation

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 generates an SQL script that you can provide 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 need them later during the domain creation process.

2. Select **AS Common Schemas**.

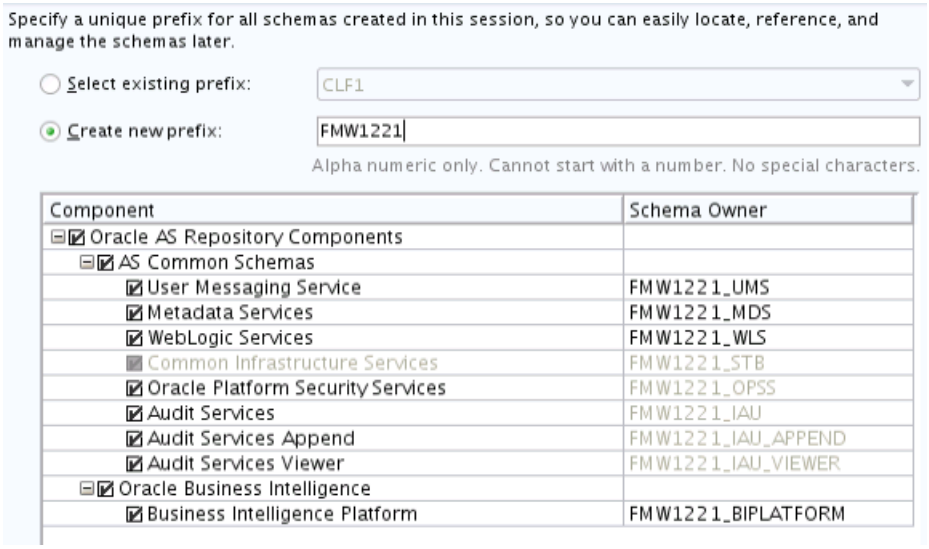
   When you select **AS Common Schemas**, all of the schemas in this section are automatically selected.

   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*.

Figure 10-1 Creating Custom Prefix



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:**
>
> Make a note of the passwords you set on this screen; you need them later 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.
See About the Repository Creation Utility in *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.
When you reach the Completion Summary screen, click **Close** to close the RCU.

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.

- Starting the Configuration Wizard
- Navigating the Configuration Wizard Screens to Configure the BI Domain

Starting the Configuration Wizard

To begin domain configuration, run the following command in the Oracle Fusion Middleware Oracle home on BIHOST1.

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

Navigating the Configuration Wizard Screens to Configure the BI Domain

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 – 12.2.1.3.0 [bi]**

  Selecting this template automatically selects the following dependencies:

  - Oracle MapViewer – 12.2.1.3.0 [oracle_common]
  - Oracle Enterprise Manager – 12.2.1.3.0 [em]
  - Oracle WSM Policy Manager – 12.2.1.3.0 [oracle_common]
  - Oracle JRF - 12.2.1.3.0 [oracle_common]
  - WebLogic Coherence Cluster Extension - 12.2.1.3.0 [wlserver]
- **Oracle BI Publisher Suite – 12.2.1.3.0 [bi]**
- **Oracle BI Essbase Suite – 12.2.1.3.0 [bi]**

In addition, the **Basic WebLogic Server Domain – 12.2.1.3.0 [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 High Availability Options
Use the High Availability Options screen to configure service migration and persistence settings that affect high availability.
The **Enable Automatic Service Migration** option allows pinned services to migrate automatically to a healthy Managed Server for failover. However, Oracle BI does not support Automatic Service Migration. Ensure that the **Enable Austomatic Service Migration** option is not selected.
The JTA Transaction Log Persistence section has two options: **Default Persistent Store** and **JDBC TLog Store**. Oracle recommends that you select **JDBC TLog Store**. You use this option to configure a component to use JDBC stores for all its JMS servers. When you complete the configuration, you have a cluster and JDBC persistent stores are set up for Transaction logs.
For more details on persistent and TLOG stores, see:

- Using the Default Persistent Store
- Using a JDBC TLOG Store

Set **JMS Server Persistence** to **JMS JDBC Store**.
A persistent **JMS store** is a physical repository for storing persistent message data and durable subscribers. It can be either a disk-based **file store** or a JDBC-

accessible database. You can use a **JMS file store** for paging of messages to disk when memory is exhausted.

- JMS File Store — Configures a component to use JMS File Stores.
- JMS JDBC Store — Configures a component to use JDBC stores for all its JMS servers. When you complete the configuration, you have a cluster and JDBC persistent stores are configured for the JMS servers.

Select the **File Store** Modify Settings option in the Advanced Configuration screen to change settings. In the File Stores screen, you can set file store names, directories and synchronous write policies.

Task 4 Selecting the Application Home Location
On the Application Location screen, specify the value of the *APPLICATION_HOME* variable, as defined in File System and Directory Variables Used in This Guide.

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

Task 5 Configuring the Administrator Account
On the Administrator Account screen, specify the user name and password for the default WebLogic Administrator account for the domain.
Make a note of the user name and password specified on this screen; you will need these credentials later to boot and connect to the domain's Administration Server.

Task 6 Specifying the Domain Mode and JDK
On the Domain Mode and JDK screen:

- Select **Production** in the Domain Mode field.
- Select the **Oracle Hotspot** JDK in the JDK field.

Selecting **Production Mode** on this screen gives your environment a higher degree of security, requiring a user name and password to deploy applications and to start the Administration Server.

> **Tip:**
>
> More information about the options on this screen, including the differences between development mode and production mode, can be found in Domain Mode and JDK in *Creating WebLogic Domains Using the Configuration Wizard*.
> In production mode, a boot identity file can be created to bypass the need to provide a user name and password when starting the Administration Server. See Creating the boot.properties File.

Task 7 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:

Field	Description
DBMS/Service	Enter the service name for the Oracle RAC database where you will install the product schemas. For example: `orcl.example.com` Be sure 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. 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 Schema Password	Enter the user name and password for connecting to the database's Service Table schema. This is the schema user name and password that was specified for the Service Table component on the Schema Passwords screen in RCU. The default user name is *prefix*_STB, where *prefix* is the custom prefix that you defined in RCU.

Click **Get RCU Configuration** when you are finished specifying the database connection information. The following output in the Connection Result Log indicates that the 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 8 Specifying JDBC Component Schema Information
Verify that the values on the JDBC Component Schema screen are correct for all schemas.
The schema table should be populated because you selected **Get RCU Data** on the previous screen. As a result, the Configuration Wizard locates the database connection values for all the schemas required for this domain.
At this point, the values are configured to connect to a single-instance database. However, for an enterprise deployment, you should use a highly available Real Application Clusters (RAC) database, as described in Preparing the Database for an Enterprise Deployment.
In addition, Oracle recommends that you use an Active GridLink datasource for each of the component schemas. For more information about the advantages of using GridLink data sources to connect to a RAC database, see Database Considerations in the *High Availability Guide*.
To convert the data sources to GridLink:

1. Select all the schemas by selecting the check box in the first header row of the schema table.
2. Click **Convert to GridLink** and click **Next**.

Task 9 Providing the GridLink Oracle RAC Database Connection Details
On the GridLink Oracle RAC Component Schema screen, provide the information required to connect to the RAC database and component schemas, as shown in Table 10-4.

Element	Description and Recommended Value
SCAN, Host Name, and Port	Select the **SCAN** check box. In the **Host Name** field, enter the Single Client Access Name (SCAN) Address for the Oracle RAC database. In the **Port** field, enter the SCAN listening port for the database (for example, 1521)
ONS Host and Port	In the **ONS Host** field, enter the SCAN address for the Oracle RAC database. In the Port field, enter the ONS Remote port (typically, 6200). To obtain the ONS information on the GridLink with Oracle RAC Database, check the ons.config file on either nodes of the RAC machine. The ons.config file is present at the following location: *GRID_HOME*/opmn/conf/ons.config. For example, /u01/app/12.2.1.x/grid/opmn/conf/ons.config.

Element	Description and Recommended Value
Enable Fan	Select the **Enable Fan** check box to receive and process FAN events,

For more information about specifying the information on this screen, as well as information about how to identify the correct SCAN address, see Configuring Active GridLink Data Sources with Oracle RAC in *High Availability Guide*.
You can also click **Help** to display a brief description of each field on the screen.

Task 10 Testing the JDBC Connections
Use the JDBC Component Schema Test screen to test the data source connections you have just configured.
A green check mark in the Status column indicates a successful test. If you encounter any issues, see the error message in the Connection Result Log section of the screen, fix the problem, then try to test the connection again.

> **Tip:**
>
> More information about the other options on this screen can be found in Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*

Task 11 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.

> **Note:**
>
> The `jms.queue.auth` user must be created with default authenticator after starting the Administration Server and before starting the Managed Server/ system components.

Task 12 Selecting Advanced Configuration
To complete domain configuration for the topology, select the following options on the Advanced Configuration screen:

- **Administration Server**

  This is required to properly configure the listen address of the Administration Server.

- **Node Manager**

  This is required to configure Node Manager.

- **Topology**

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

- **File Store**

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

  Do not select this option if you have selected JDBC persistent store.

> **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 13 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 14 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type.
Under **Node Manager Credentials**, specify the username and the password same as that of the admin user.

> **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 15 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	BIHOST1VHN	7003	No	Disabled	BISUITE-MAN-SVR

Task 16 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.
3. From the Dynamic Server Groups drop-down list, select **Unspecified**.

> **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 17 Assigning Server Templates
Click **Next** to continue.

Task 18 Configuring Dynamic Servers
Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

1. Confirm that the **Dynamic Cluster**, **Calculated Listen Port**, and **Calculated Machine Names** checkboxes on this screen are unchecked.
2. Confirm the **Server Template** selection is **Unspecified**.
3. Click **Next**.

Task 19 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 20 Configuring Coherence Clusters
Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain.
In the **Cluster Listen Port**, enter `9991`.

> **Note:**
>
> For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information*.

Task 21 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.

   Specify the values listed in the following table to define the Name and Node Manager Listen Address of each machine.

   > **Note:**
   >
   > Do not specify `localhost` in the Node Manager Listen Address field.

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`.	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. For the BIHOST1 use the port 5556.

Note:

If BIHOSTT1 and ADMINHOSTT are rerunni

Name	Node Manager Listen Address	Node Manager Listen Port

ngonthesameserver,theneachoftheirnodemanagersmust

Name	Node Manager Listen Address	Node Manager Listen Port

runondifferentports,i.e.port5556forBIHOST1andport

Name	Node Manager Listen Address	Node Manager Listen Port
		5557 for ADMINHOST.
ADMINHOST	Enter the value of the ADMINVHN variable.	5556

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

Task 22 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 23 Creating Virtual Targets
Click **Next** to proceed to the next screen.

Task 24 Creating Partitions
Click **Next** to proceed to the next screen.

Task 25 Configuring the JMS File Store

> **Note:**
>
> The Configuring the JMS File Store screen does not appear if you have selected the JDBC for the JMS File Store.

When you configure a domain using the BI configuration template, you should select the proper location of the Metadata Services (MDS) JMS File Store, especially when you are configuring an enterprise deployment.
On 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 26 Reviewing Your Configuration Specifications and Configuring the Domain
The Configuration Summary screen contains the detailed configuration information for the domain you are about to create. Review the details of each item on the screen and verify that the information is correct.
You can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation 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 27 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 close the Configuration Wizard.

Disabling the Derby Database

Disable the embedded Derby database, which is a file-based database, packaged with Oracle WebLogic Server. The Derby database is used primarily for development environments. As a result, you must disable it when you are configuring a production-ready enterprise deployment environment; otherwise, the Derby database process starts automatically when you start the Managed Servers.

To disable the Derby database:

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

   ```
   cd WL_HOME/common/derby/lib
   ```

2. Rename the Derby library jar file:

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

3. If each host uses a separate file system, repeat steps 1 and 2 on each host.

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',po
   rtMonitor='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',por
   tMonitor='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()
```

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 by entering the following commands:

```
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. 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. To configure the SampleAppLite application, create a script with a filename such as SampleAppLite_config.txt and insert the following lines.

   Substitute the proper path for both the *ORACLE_HOME* and the *ASERVER_HOME* variables.

   ```
   import os, sys
   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()
   srcZip='ORACLE_HOME/bi/bifoundation/samples/sampleapplite/SampleAppLite-
   datafiles.zip'
   destDir='ASERVER_HOME/bidata/service_instances/service1/data'
   unzip(srcZip,destDir)
   ```

8. Run the script in WLST by entering the following command:

```
wls:/offline> execfile('SampleAppLite_config.txt')
wls:/offline> exit()
```

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.

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.

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
```

After configuring CSS, you must update the jps-config.xml file present at the following locations:

Primary host - BIHOST1:

```
/u01/oracle/config/domains/biedg_domain/config/fmwconfig/jps-config.xml
```

```
/u02/oracle/config/domains/biedg_domain/config/fmwconfig/jps-config.xml
```

Secondary host - BIHOST2:

```
/u01/oracle/config/domains/biedg_domain/config/fmwconfig/jps-config.xml
```

```
/u02/oracle/config/domains/biedg_domain/config/fmwconfig/jps-config.xml
```

Add the following in to the jps-config.xml file, if not present already:

```
<property name="odbc.dsn" value="opss_datasource"/>
```

The jps-config.xml file should look like this:

```
<propertySet name="props.db.1">
            <property name="server.type" value="DB_ORACLE"/>
            <property name="oracle.security.jps.farm.name"
value="cn=opssSecurityStore"/>
```

```
            <property name="datasource.jndi.name" value="jdbc/OpssDataSource"/>
            <property name="oracle.security.jps.db.useDSAdminMapKey"
value="true"/>
            <property name="oracle.security.jps.ldap.root.name"
value="cn=opssRoot"/>
            .
            .
            .
            <property name="odbc.dsn" value="opss_datasource"/>
</propertySet>
```

After you update the jps-config.xml file, restart BIHOST1 and BIHOST2 using stop.sh and start.sh commands.

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.

- 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 to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.
- Starting the Administration Server Using the Node Manager
  Use these steps to start the Administration Server using the Node Manager.
- Validating the Administration Server
  Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control; both of these are installed and configured on the Administration Servers.
- Creating a Separate Domain Directory for Managed Servers on BIHOST1
  When you initially create the domain for enterprise deployment, the domain directory resides on a shared disk. This default domain directory is used to run the Administration Server. You can now create a copy of the domain on the local storage for both BIHOST1 and BIHOST2. The domain directory on the local (or private) storage is used to run the Managed Servers.
- Starting the Node Manager in the Managed Server Domain Directory on 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.

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 nodemanager.properties file for editing:

      ASERVER_HOME`/nodemanager/nodemanager.properties`

   b. Make sure the `ListenAddress` property is set to the value of the ADMINVHN virtual IP address.
   c. Make sure that QuitEnabled is set to 'true'. If this line is not present in the nodemanager.properties file, add the following line:

      ```
      QuitEnabled=true
      ```

2. Change to the following directory:

   ASERVER_HOME`/bin`

3. Start the Node Manager by entering the following command:

   ```
   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*.

Creating the boot.properties File

You must create a `boot.properties` if you want to start the Administrator Server without being prompted for the Administrator Server credentials. This step is required in an enterprise deployment. When you start the Administration Server, the credentials that you enter in this file are encrypted.

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

1. Create the following directory structure:

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

2. In a text editor, create a file called `boot.properties` in the `security` directory that you created in the previous step, and enter the Administration Server credentials that you defined when you ran the Configuration Wizard to create the domain:

   ```
   username=adminuser
   password=password
   ```

> **Note:**
>
> When you start the Administration Server, the `username` and `password` entries in the file are encrypted.
>
> For security reasons, minimize the amount of time the entries in the file are left unencrypted; after you edit the file, you should start the server as soon as possible so that the entries are encrypted.

3. Save the file and close the editor.

Starting the Administration Server Using the Node Manager

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()
```

Validating the Administration Server

Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Server Administration Console and Oracle Enterprise Manager Fusion Middleware Control; both of these are installed and configured on the Administration Servers.

To navigate to 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

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 is 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 is used to run the Managed Servers.

Placing the MSERVER_HOME on local storage is recommended to eliminate the potential contention and overhead caused by servers writing logs to shared storage. It is also faster to load classes and jars need from the domain directory, so any temporary or cache data that the Managed Servers use from the domain directory is processed quicker.

As described in Preparing the File System for an Enterprise Deployment, the path to the Administration Server domain home is represented by the 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. Sign 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=/full_path/bidomaintemplate.jar \
          -template_name=bi_domain_template \
      -log_priority=DEBUG \
          -log=/tmp/pack.log
```

   In this example:

   - Replace *ASERVER_HOME* with the actual path to the domain directory you created on the shared storage device.
   - Replace *full_path* with the complete path to the location where you want to create the domain template jar file. You need to reference this location when you copy or unpack the domain template jar file. It is recommended to choose a shared volume other than *ORACLE_HOME*, or write to `/tmp/` and copy the files manually between servers.

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

     SHARED_CONFIG_DIR/domains/template_filename.jar
   - The `bidomaintemplate.jar` file is a sample name for the jar file that you create, which contains the domain configuration files.

- The `bi_domain_template` label is the label is assigned to the template data stored in the template file.

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

> **Tip:**
>
> For more information about the `pack` and `unpack` commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

3. If you have not already, create the recommended directory structure for the Managed Server domain on the 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=/full_path/bidomaintemplate.jar \
        -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME
```

> **Note:**
>
> The `-overwrite_domain` option in the `unpack` command allows you to unpack a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this unpack operation.
>
> Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional JAVA command-line options for running the servers, or specify additional environment variables. Any customizations that you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when you use the `pack` and `unpack` commands.

In this example:

- Replace *MSERVER_HOME* with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain is unpacked.

- Replace `/full_path/bidomaintemplate.jar` with the complete path and file name of the domain template jar file that you created when you ran the pack command to pack the domain on the shared storage device.
- Replace *APPLICATION_HOME* with the complete path to the Application directory for the domain on shared storage. See File System and Directory Variables Used in This Guide.

> **Tip:**
>
> For more information about the `pack` and `unpack` commands, see Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*.

5. Change directory to the newly created Managed Server directory and verify that the domain configuration files were copied to the correct location on the BIHOST1 local storage device.

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.

> **Note:**
>
> The Node Manager for the Managed Server's *MSERVER_HOME* will be reset every time the domain configuration is unpacked. The `ListenAddress` will be changed to the *ADMINVHN* instead of the correct hostname. This needs to be changed to the correct value before starting the Node Manager service after an unpack is performed.

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

1. Verify that the listen address in the nodemanager.properties file is set correctly, by completing the following steps:
   a. Change to the following directory:

      MSERVER_HOME`/nodemanager/`

   b. Open the nodemanager.properties file for editing.
   c. Update the `ListenAddress` property to the correct hostname as follows:

      ```
      BIHOST1: ListenAddress=BIHOST1
      ```

d. Update the `ListenPort` property with the correct Listen Port details.

e. Make sure that QuitEnabled is set to 'true'. If this line is not present in the nodemanager.properties file, add the following line:

```
QuitEnabled=true
```

2. Change to the following directory:

```
MSERVER_HOME/bin
```

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

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

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

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:

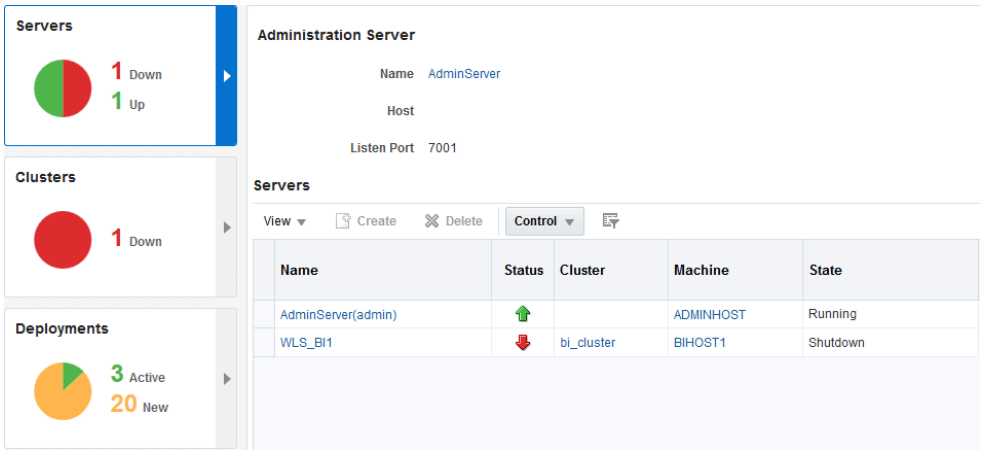
- Replace *ADMINVHN* with the host name assigned to the ADMINVHN Virtual IP address.
- Port 7001 is the typical port used for the Administration Server console and Fusion Middleware Control. However, you should use the actual URL that was displayed at the end of the Configuration Wizard session when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

Figure 10-2 Viewing the Status of Managed Servers



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

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:

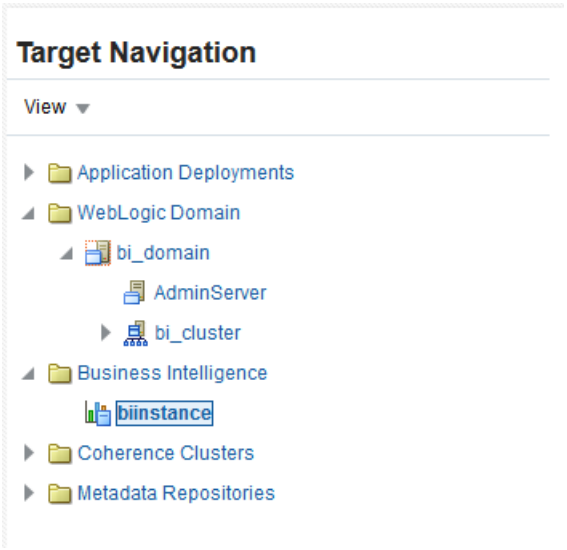
   `http://ADMINVHN:7001/em`

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**.

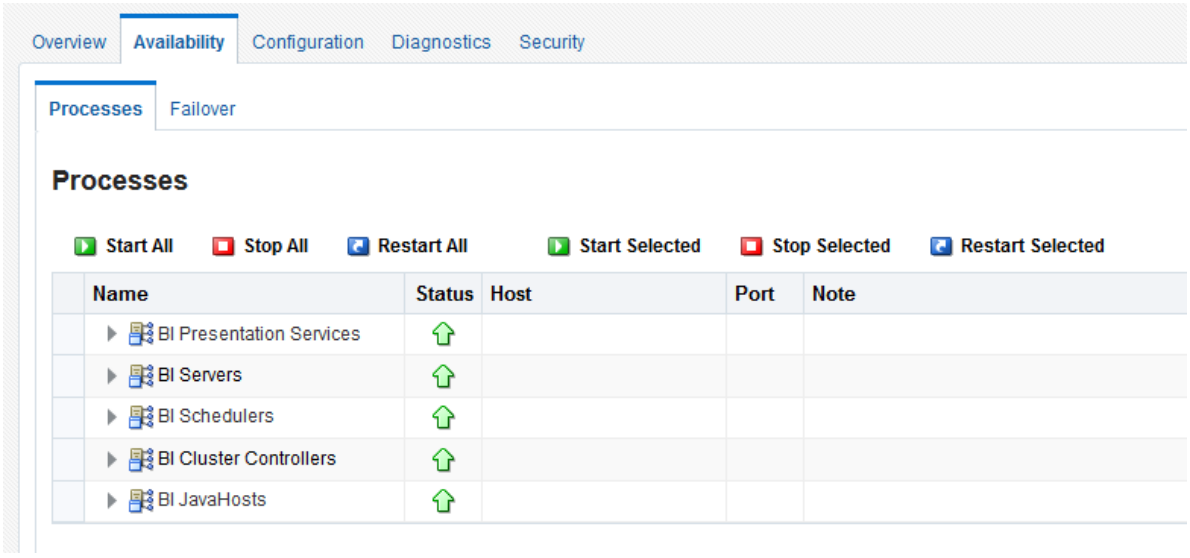
Figure 10-3 Selecting biinstance from the Target Navigation Pane



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.

Figure 10-4 Starting All Components



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.

See About the Global Cache in *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:

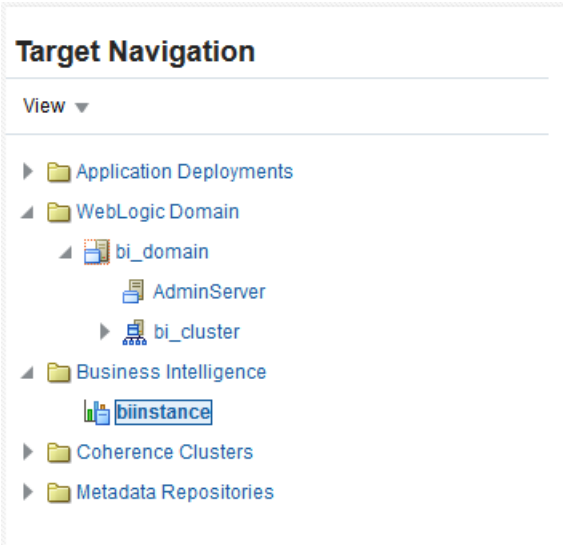
      ```
      http://ADMINVHN:7001/em
      ```

   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**.

      Figure 10-5 Selecting biinstance from the Target Navigation Pane

      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.

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://BIHOST1VHN: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://BIHOST1VHN: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://BIHOST1VHN:7003/aps/Essbase`

Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group

When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server authentication provider (`DefaultAuthenticator`). However, for an enterprise deployment, Oracle recommends that you use a dedicated, centralized LDAP-compliant authentication provider.

The following topics describe how to use the Oracle WebLogic Server Administration Console to create a new authentication provider for the enterprise deployment domain. This procedure assumes that you have already installed and configured a supported LDAP directory, such as Oracle Unified Directory or Oracle Internet Directory.

- 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 Administration Role to the New Administration Group

- Adding weblogic_bi User to the BIServiceAdministrator Role
- Updating the boot.properties File and Restarting the System
- Adding the wsm-pm Role to the Administrators Group
  After you configure a new LDAP-based Authorization Provider and restart the Administration Server, add the enterprise deployment administration LDAP group (`BIAdministrators`) as a member to the `policy.Updater` role in the `wsm-pm` application stripe.

About the Supported Authentication Providers

Oracle Fusion Middleware supports a variety of LDAP authentication providers. See Identity Store Types and WebLogic Authenticators in *Securing Applications with Oracle Platform Security Services*.

The instructions in this guide assume that you are using one of the following providers:

- Oracle Unified Directory
- Oracle Internet Directory
- Microsoft Active Directory

> **Note:**
>
> By default, the instructions here describe how to configure the identity service instance to support querying against a single LDAP identity store with an unencrypted connection.
>
> If the connection to your identity provider has to be secured through SSL, then additional keystone configuration is required for role management in the Enterprise Manager Fusion Middleware Control to function correctly. For additional configuration information, see Doc ID 1670789.1 at support.oracle.com.
>
> Also, you can configure the service to support a virtualized identity store, which queries multiple LDAP identity stores, by 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*.

About the Enterprise Deployment Users and Groups

The following topics provide important information on the purpose and characteristics of the enterprise deployment administration users and groups.

- About Using Unique Administration Users for Each Domain
- About the Domain Connector User
- About Adding Users to the Central LDAP Directory
- About Product-Specific Roles and Groups for Oracle Business Intelligence
- Example Users and Groups Used in This Guide

About Using Unique Administration Users for Each Domain

When you use a central LDAP user store, you can provision users and groups for use with multiple Oracle WebLogic Server domains. As a result, there is a possibility that one WebLogic administration user can have access to all the domains within an enterprise.

It is a best practice to create and assign a unique distinguished name (DN) within the directory tree for the users and groups that you provision for the administration of your Oracle Fusion Middleware domains.

For example, 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 **`BIAdministrators`**.

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 Governance do not reconcile this user.

About Adding Users to the Central LDAP Directory

After you configure a central LDAP directory to be the authenticator for the enterprise domain, then you should add all new users to the new authenticator and not to the default WebLogic Server authenticator.

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.

To enable virtualization:

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

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

2. Find the following section:

   ```
   <serviceInstance name="idstore.ldap" provider="idstore.ldap.provider">
   ```

3. Add the following line under the `serviceInstance` section or update the virtualize 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*.

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 `BIAdministrators` group. After they are added to the `BIAdministrators` group, each product administrator user can administer the domain with the same set of privileges for performing administration tasks.

For instructions on adding additional roles to the `BIAdministrators` group, see Common Configuration and Management Tasks for an Enterprise Deployment.

Example Users and Groups Used in This Guide

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

- Admin User DN:

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

- Admin Group DN:

  ```
  cn=BIAdministrators,cn=groups,dc=example,dc=com
  ```

- Product-specific LDAP Connector User:

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

  This is the user that you use to connect WebLogic Managed Servers to the LDAP authentication provider. This user must have permissions to read and write to the Directory Trees:

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

> **Note:**
>
> This user needs to be granted membership in the following groups to provide read and write access:
>
> ```
> cn=orclFAUserReadPrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAUserWritePrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAGroupReadPrivilegeGroup,cn=groups,dc=example,dc=com
> cn=orclFAGroupWritePrivilegeGroup,cn=groups,dc=example,dc=com
> ```

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:

ASERVER_HOME/servers/AdminServer/security

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 following contents 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 use 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=example,dc=com
> changetype: modify
> add: uniquemember
> uniquemember: cn=biLDAP,cn=systemids,dc=example,dc=com
>
> dn:
> cn=orclFAGroupReadPrivilegeGroup,cn=groups,dc=example,dc=com
> changetype: modify
> add: uniquemember
> uniquemember: cn=biLDAP,cn=systemids,dc=example,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 idstore.example.com
                                    -D "cn=oudadmin" \
                                    -w password \
                                    -p 1389 \
                                    -f domain_user.ldif
   ```

   For Oracle Internet Directory:

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

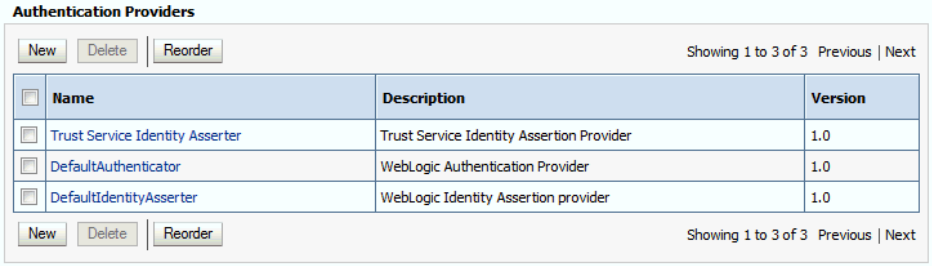
Creating the New Authentication Provider

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.

Figure 10-6 List of Available Authentication Providers



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 persist the change of the control flag setting.

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

    Note that only the required fields are discussed in this procedure. For information about all the fields on this page, consider the following resources:

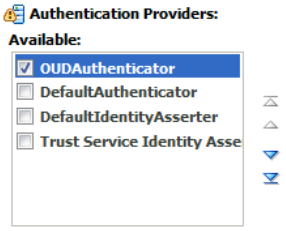
    - To display a description of each field, click **Help** on the **Provider Specific** tab.
    - For more information on setting the **User Base DN**, **User From Name Filter**, and **User Attribute** fields, see Configuring Users and Groups in the Oracle Internet Directory and Oracle Virtual Directory Authentication Providers in *Administering Security for Oracle WebLogic Server*.

Parameter	Sample Value	Value Description
Host	For example: `idstore.example.com`	The LDAP server's server ID.
Port	For example: `1389`	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.
All Users Filter	`(&(uid=*)(objectclass=person))`	Instead of a default search criteria for **All Users Filter**, search all users based on the `uid` value. If the **User Name Attribute** for the user object class in the LDAP directory structure is a type other than `uid`, then change that type in the **User From Name Filter** field. For example, if the **User Name Attribute** type is `cn`, then this field should be set to: `(&(cn=*)(objectclass=person))`
User From Name Filter	For example: `(&(uid=%u)(objectclass=person))`	If the **User Name Attribute** for the user object class in the LDAP directory structure is a type other than `uid`, then change that type in the settings for the **User From Name** Filter. For example, if the **User Name Attribute** type is `cn`, then this field should be set to: `(&(cn=%u)(objectclass=person))`.
User Name Attribute	For example: `uid`	The attribute of an LDAP user object that specifies the name of the user.
Use Retrieved User Name as Principal	Checked	Must be turned on.
Group Base DN	For example: `cn=groups,dc=example,dc=com`	Specify the DN that points to your Groups node.

Parameter	Sample Value	Value Description
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.

Figure 10-7 Reordering the Authentication Providers



17. Click **OK**.
18. On the Providers Page, click **DefaultAuthenticator**.
19. From the Control Flag drop-down, select **SUFFICIENT**.
20. Click **Save** to update the DefaultAuthenticator settings.
21. In the Change Center, click **Activate Changes**.
22. 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.

23. 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.

24. 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.

Provisioning an Enterprise Deployment Administration User and Group

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

To provision the administration user and group in LDAP provider:

1. Create an `LDIF` file named `admin_user.ldif` with the following contents 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 idstore.example.com
```

```
                                    -D "cn=oudadmin" \
                                    -w password \
                                    -p 1389 \
                                    -f admin_user.ldif
```

For Oracle Internet Directory:

```
OID_ORACLE_HOME/bin/ldapadd -h idstore.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 following contents and then save the file:

```
dn: cn=BIAdministrators,cn=Groups,dc=example,dc=com
displayname: BIAdministrators
objectclass: top
objectclass: groupOfUniqueNames
objectclass: orclGroup
uniquemember: cn=weblogic_bi,cn=users,dc=example,dc=com
cn: BIAdministrators
description: Administrators Group for the Oracle Business Intelligence
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 1380 \
                                 -f admin_group.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_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 that you provisioned are listed on the page.

Adding the Administration Role to the New Administration Group

After you add 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 by using the administration credentials that you provided in the Configuration Wizard.

   Do not use the credentials for the administration user that you created and provided for the new authentication provider.
2. In the left pane of the Administration Console, click **Security Realms**.
3. Click the default security realm (**myrealm**).
4. Click the **Roles and Policies** tab.
5. Expand the **Global Roles** entry in the table and click **Roles**.

Figure 10-8 Global Roles Under Security Realms

6. Click the **Admin** role.

Figure 10-9 Adding Conditions for the Admin Role

7. Click **Add conditions**.
8. Select **Group** from the **Predicate List** drop-down menu, and then click **Next**.
9. Enter `BIAdministrators` in the **Group Argument Name** field, and then click **Add**.

   `BIAdministrators` is added to the list box of arguments.
10. Click **Finish** to return to the Edit Global Role page.

    The `BIAdministrators` group is now listed.
11. Click **Save** to finish adding the **Admin** Role to the `BIAdministrators` group.
12. Validate that the changes were made by logging in to the WebLogic Administration Server Console by 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 that

you just provisioned in the new authentication provider, then you have configured the provider successfully.

Adding weblogic_bi User to the BIServiceAdministrator Role

To add the weblogic_bi user to the BIServiceAdministrator role:

1. Sign in to the Fusion Middleware Control with the Administrator credentials.
2. From the WebLogic Domain menu, select **Security**, and then click **Application Roles**.
3. Select **obi** as the application stripe.
4. Edit the `BIServiceAdministrator` role.

   Add `weblogic_bi` user as a member to this role.
5. Go to the **Members** section and click the **+** (Add) icon.
6. In the **Search** section, select **Type** as **User**.
7. In the **Advanced Options** section, select **Check to enter principal name here instead of searching from above**.
8. Add `weblogic_bi` user after selecting the type as **User**.
9. Click **OK** on the Edit Application Role page.

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.

Adding the wsm-pm Role to the Administrators Group

After you configure a new LDAP-based Authorization Provider and restart the Administration Server, add the enterprise deployment administration LDAP group

(BIAdministrators) as a member to the policy.Updater role in the wsm-pm application stripe.

> **Note:**
>
> This is not required for Access infrastructure.

1. Use the Oracle WebLogic Server Administration Server credentials to log in to Oracle Enterprise Manager Fusion Middleware Control.

   These are the credentials you created when you initially configured the domain and created the Oracle WebLogic Server Administration user name (typically, weblogic_bi) and password.
2. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.
3. Select the **wsm-pm** application stripe from the Application Stripe drop-down menu.
4. Click the triangular play icon next to the role name text box to search for all role names in the wsm-pm application stripe.
5. Select the row for the **policy.Updater** role to be edited.
6. Click the Edit icon to edit the role.
7. Click the Add icon on the Edit Application Role page.
8. In the Add Principal dialog box, select **Group** from the **Type** drop-down menu.
9. Search for the enterprise deployment administrators group, by entering the group name BIAdministrators in the **Principal Name Starts With** field and clicking the right arrow to start the search.
10. Select the appropriate administrators group in the search results and click **OK**.
11. Click **OK** on the Edit Application Role page.

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.

See Performing Backups and Recoveries for an Enterprise Deployment.

11

Configuring Oracle HTTP Server for an Enterprise Deployment

When you configure the web tier, you have the option of using Oracle HTTP Server or Oracle Traffic Director. If you choose to use Oracle HTTP Server, then you must install Oracle HTTP Server on each of the web tier hosts and configure Oracle HTTP standalone domains on each host.

The Oracle HTTP Server instances on the web tier direct HTTP requests from the hardware load balancer to specific Managed Servers in the application tier.

Before you configure Oracle HTTP Server, be sure to review Understanding the Web Tier.

> **Note:**
>
> If you plan to configure Oracle Managed File Transfer, then you must configure Oracle Traffic Director to route FTP and SFTP requests over TCP.

- Variables Used When Configuring the Oracle HTTP Server
  As you perform the tasks in this chapter, you reference the directory variables that are listed in this topic.
- About the Oracle HTTP Server Domains
  In an enterprise deployment, each Oracle HTTP Server instance is configured on a separate host and in its own standalone domain. This allows for a simple configuration that requires a minimum amount of configuration and a minimum amount of resources to run and maintain.
- Installing a Supported JDK
- Installing Oracle HTTP Server on WEBHOST1
  It is important to understand the procedure for installing the Oracle HTTP Server software on the web tier.
- Creating an Oracle HTTP Server Domain on WEBHOST1
  The following topics describe how to create a new Oracle HTTP Server standalone domain on the first web tier host.
- Installing and Configuring an Oracle HTTP Server Domain on WEBHOST2
  After you install Oracle HTTP Server and configure a domain on WEBHOST1, then you must also perform the same tasks on WEBHOST2.
- Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2
  It is important to understand how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.

- Configuring Oracle HTTP Server to Route Requests to the Application Tier
  It is important to understand how to update the Oracle HTTP Server configuration files so that 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 you successfully configure a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

Variables Used When Configuring the Oracle HTTP Server

As you perform the tasks in this chapter, you reference the directory variables that are listed in this topic.

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

- *WEB_ORACLE_HOME*
- *WEB_DOMAIN_HOME*
- *JAVA _HOME*

In addition, you reference the following virtual IP (VIP) address and host names:

- ADMINVHN
- WEBHOST1
- WEBHOST2

About the Oracle HTTP Server Domains

In an enterprise deployment, each Oracle HTTP Server instance is configured on a separate host and in its own standalone domain. This allows for a simple configuration that requires a minimum amount of configuration and a minimum amount of resources to run and maintain.

> **Note:**
>
> Oracle Fusion Middleware requires that a certified Java Development Kit (JDK) is installed on your system and JAVA_HOME is set on the web tier hosts.

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

Installing a Supported JDK

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

- Locating and Downloading the JDK Software

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

Locating and Downloading the JDK Software

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

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

```
http://www.oracle.com/technetwork/java/index.html
```

Be sure to navigate to the download for the Java SE JDK.

Installing the JDK Software

Oracle Fusion Middleware requires you to install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the following locations:

On the local storage device for each of the Web tier host computers. The Web tier host computers, which reside in the DMZ, do not necessarily have access to the shared storage on the application tier.

See the Understanding the Recommended Directory Structure for an Enterprise Deployment.

To install JDK 1.8.0_131:

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

   ```
   cd download_dir
   ```

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

   ```
   tar -xzvf jdk-8u131-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_131 /u02/oracle/products/jdk
   ```

   See File System and Directory Variables Used in This Guide.

4. Define the *JAVA_HOME* and *PATH* environment variables for running Java on the host computer.

For example:

```
export JAVA_HOME=/u02/oracle/products/jdk
export PATH=$JAVA_HOME/bin:$PATH
```

5. Run the following command to verify that the appropriate `java` executable is in the path and your environment variables are set correctly:

   ```
   java -verison
   ```

   The Java version in the output should be displayed as `1.8.0_131`.

Installing Oracle HTTP Server on WEBHOST1

It is important to understand the procedure for installing the Oracle HTTP Server software on the web tier.

- Starting the Installer on WEBHOST1
- Navigating the Oracle HTTP Server Installation Screens
- Verifying the Oracle HTTP Server Installation

Starting the Installer on WEBHOST1

To start the installation program, perform the following steps.

1. Log in to WEBHOST1.
2. Go to the directory in which you downloaded the installation program.
3. Launch the installation program by entering the following command:

   ```
   ./fmw_12.2.1.1.0_ohs_linux64.bin
   ```

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

Navigating the Oracle HTTP Server Installation Screens

The following table lists the screens in the order that the installation program displays them.

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

Table 11-1 Oracle HTTP Server Installation Screens

Screen	Description
Installation Inventory Setup	On UNIX operating systems, this screen appears if you install any Oracle product on this host for the first time. Specify the location where you want to create your central inventory. Ensure that the operating system group name selected on this screen has write permissions to the central inventory location. See Understanding the Oracle Central Inventory in *Installing Software with the Oracle Universal Installer*. **Note:** Oracle recommends that you configure the central inventory directory within the products directory. Example: `/u02/oracle/products/oraInventory` You may also need to execute the `createCentralInventory.sh` script as root from the `oraInventory` folder after the installer completes.
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search the local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For the purposes of an enterprise deployment, enter the value of the *WEB_ORACLE_HOME* variable listed in Table 7-3.
Installation Type	Select **Standalone HTTP Server (Managed independently of WebLogic server)**. This installation type allows you to configure the Oracle HTTP Server instances independently from any other existing Oracle WebLogic Server domains.
JDK Selection	For the value of JDK Home, enter the value of *JAVA_HOME* that you set when installing the JDK software.

Table 11-1 (Cont.) Oracle HTTP Server Installation Screens

Screen	Description
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, verify that your host computers and the required software meet the system requirements and certification information described in Host Computer Hardware Requirements and Operating System Requirements for the Enterprise Deployment Topology.
Installation Summary	Use this screen to verify the installation options that you selected. If you want to save these options to a response file, click **Save Response File** and provide the location and name of the response file. Response files can be used later in a silent installation situation. See Using the Oracle Universal Installer in Silent Mode in *Installing Software with the Oracle Universal Installer*.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click **Finish** to close the installer.

Verifying the Oracle HTTP Server Installation

To verify that the Oracle HTTP Server installation completed successfully, use the `ls --format=single-column` command to list the files that are 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
slax
sqlplus
srvm
webgate
wlserver
xdk
```

Creating an Oracle HTTP Server Domain on WEBHOST1

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

- Starting the Configuration Wizard on WEBHOST1
- Navigating the Configuration Wizard Screens for an Oracle HTTP Server Domain

Starting the Configuration Wizard on WEBHOST1

To start the Configuration Wizard, navigate to the following directory and start the WebLogic Server Configuration Wizard, as follows:

```
cd WEB_ORACLE_HOME/oracle_common/common/bin
./config.sh
```

Navigating the Configuration Wizard Screens for an Oracle HTTP Server Domain

Oracle recommends that you create a standalone domain for the Oracle HTTP Server instances on each web tier host.

The following topics describe how to create a new standalone Oracle HTTP Server domain:

- Task 1, Selecting the Domain Type and Domain Home Location
- Task 2, Selecting the Configuration Templates
- Task 3, Selecting the JDK for the Web Tier Domain.
- Task 4, Configuring System Components
- Task 5, Configuring OHS Server
- Task 7, Reviewing Your Configuration Specifications and Configuring the Domain
- Task 8, Writing Down Your Domain Home

Task 1 Selecting the Domain Type and Domain Home Location
On the Configuration Type screen, select **Create a new domain**.
In the **Domain Location** field, enter the value assigned to the *WEB_DOMAIN_HOME* variable.
Note the following:

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

Task 2 Selecting the Configuration Templates
On the Templates screen, select **Oracle HTTP Server (Standalone) - 12.2.1.3.0 [ohs]**.

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

Task 3 Selecting the JDK for the Web Tier Domain.
Select the Oracle HotSpot JDK installed in the `/u02/oracle/products/jdk` directory prior to the Oracle HTTP Server installation.

Task 4 Configuring System Components
On the System Components screen, configure one Oracle HTTP Server instance. The screen should, by default, have a single instance defined. This is the only instance that you need to create.

1. The default instance name in the **System Component** field is `ohs1`. Use this default name when you configure `WEBHOST1`.
2. Make sure that `OHS` is selected in the **Component Type** field.
3. If an application is not responding, use the **Restart Interval Seconds** field to specify the number of seconds to wait before you attempt a restart if an application is not responding.
4. Use the **Restart Delay Seconds** field to specify the number of seconds to wait between restart attempts.

Task 5 Configuring OHS Server
Use the OHS Server screen to configure the OHS servers in your domain:

1. Select **ohs1** from the **System Component** drop-down menu.
2. In the **Listen Address** field, enter *`WEBHOST1`*.

   All the remaining fields are prepopulated, but you can change the values as required for your organization. See OHS Server in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.
3. In the **Server Name** field, verify the value of the listen address and listen port.

   It should appear as follows:

   `http://`*`WEBHOST1`*`:7777`

Task 6 Configuring Node Manager
Select **Per Domain Default Location** as the Node Manager type, and specify the user name and password for the Node Manager.

> **Note:**
>
> For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.
> For information about Node Manager configuration, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Task 7 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains detailed configuration information for the domain that you are about to create. Review the details of each item on the screen and verify that the information is correct.
If you need to make any changes, you can go back to any previous screen either by using the **Back** button or by selecting the screen in the navigation pane.
Domain creation does not begin until you click **Create**.
In the Configuration Progress screen, click **Next** when it finishes.

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

Task 8 Writing Down Your Domain Home

The Configuration Success screen shows the domain home location.
Make a note of the information provided here, as you need it to start the servers and access the Administration Server.
Click **Finish** to close the Configuration Wizard.

Installing and Configuring an Oracle HTTP Server Domain on WEBHOST2

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

1. Log in to WEBHOST2 and install Oracle HTTP Server by using the instructions in Installing Oracle HTTP Server on WEBHOST1.
2. Configure a new standalone domain on WEBHOST2 by using the instructions in Creating a Web Tier Domain on WEBHOST1.

   Use the name `ohs2` for the instance on WEBHOST2, and be sure to replace all occurrences of WEBHOST1 with WEBHOST2 and all occurrences of `ohs1` with `ohs2` in each of the examples.

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

It is important to understand how to start the Oracle HTTP Server instances on WEBHOST1 and WEBHOST2.

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

Starting the Node Manager on WEBHOST1 and WEBHOST2

Before you can start the Oracle HTTP Server instances, you must start the Node Manager on WEBHOST1 and WEBHOST2:

1. Log in to WEBHOST1 and navigate to the following directory:

   ```
   WEB_DOMAIN_HOME/bin
   ```

2. Start the Node Manager as shown in the following sections by using `nohup` and `nodemanager.out` as an example output file:

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

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

See Advanced Node Manager Configuration in *Administering Node Manager for Oracle WebLogic Server*.

Starting the Oracle HTTP Server Instances

To start the Oracle HTTP Server instances:

1. Navigate to the following directory on WEBHOST1:

   ```
   WEB_DOMAIN_HOME/bin
   ```

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

2. Enter the following command:

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

> **Note:**
>
> Every time you start the Oracle HTTP server, you will be asked for the Node Manager password. If you do not wish this behaviour, then use the following command the first time you start the Oracle HTTP server:
>
> ```
> ./startComponent.sh ohs1 storeUserConfig
> ```
>
> This time when you enter the Node Manager password, it will be encrypted and stored. Future start and stop of the Oracle HTTP server will not require you to enter the Node Manager password.

3. When prompted, enter the Node Manager password.
4. Repeat steps 1 through 3 to start the `ohs2` instance on WEBHOST2. See Starting Oracle HTTP Server Instances in *Administering Oracle HTTP Server*.

Configuring Oracle HTTP Server to Route Requests to the Application Tier

It is important to understand how to update the Oracle HTTP Server configuration files so that the web server instances route requests to the servers in the domain.

- About the Oracle HTTP Server Configuration for an Enterprise Deployment
- Modifying the httpd.conf File to Include Virtual Host Configuration Files
- Creating the Virtual Host Configuration Files for Oracle Business Intelligence
- Configuring the WebLogic Proxy Plug-In
- Validating the Virtual Server Configuration on the Load Balancer
- Validating Access to the Management Consoles and Administration Server
- Validating 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.

About the Oracle HTTP Server Configuration for an Enterprise Deployment

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

- Purpose of the Oracle HTTP Server Virtual Hosts
- Recommended Structure of the Oracle HTTP Server Configuration Files

Purpose of the Oracle HTTP Server Virtual Hosts

The reference topologies in this guide require that you define a set of virtual servers on the hardware load balancer. You can then configure Oracle HTTP Server to recognize requests to specific virtual hosts (that map to the load balancer virtual servers) by adding `<VirtualHost>` directives to the Oracle HTTP Server instance configuration files.

For each Oracle HTTP Server virtual host, you define a set of specific URLs (or context strings) that route requests from the load balancer through the Oracle HTTP Server instances to the appropriate Administration Server or Managed Server in the Oracle WebLogic Server domain.

Recommended Structure of the Oracle HTTP Server Configuration Files

Rather than adding multiple virtual host definitions to the `httpd.conf` file, Oracle recommends that you create separate, smaller, and more specific configuration files for each of the virtual servers required for the products that you are deploying. This avoids populating an already large `httpd.conf` file with additional content, and it can make troubleshooting configuration problems easier.

For example, in a typical Oracle Fusion Middleware Infrastructure domain, you can add a specific configuration file called `admin_vh.conf` that contains the virtual host definition for the Administration Server virtual host (ADMINVHN).

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

Perform the following tasks to prepare the `httpd.conf` file for the additional virtual hosts required for an enterprise topology:

1. Log in to WEBHOST1.
2. Locate the `httpd.conf` file for the first Oracle HTTP Server instance (`ohs1`) in the domain directory:

   ```
   cd WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
   ```

3. Verify if the `httpd.conf` file has the appropriate configuration as follows:
   a. Run the following command to verify the `ServerName` parameter, be sure that it is set correctly, substituting the correct value for the current WEBHOST*n*:

      ```
      grep "ServerName http" httpd.conf
      ServerName http://WEBHOST1:7777
      ```

   b. Run the following command to verify there is an include statement that includes all `*.conf` files from the moduleconf subdirectory:

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

c. If either validation fails to return results, or returns results that are commented out, open the `httpd.conf` file in a text editor and make the required changes in the appropriate locations.

```
#
# ServerName gives the name and port that the server uses to
identify itself.
# This can often be determined automatically, but we recommend you
specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP
address here.
#
ServerName http://WEBHOST1:7777
#  and at the end of the file:
# Include the admin virtual host (Proxy Virtual Host) related
configuration
include "admin.conf"
IncludeOptional "moduleconf/*.conf"
```

d. Save the `httpd.conf` file.

4. Log in to `WEBHOST2` and perform steps 2 and 3 for the `httpd.conf` file, replacing any occurrences of `WEBHOST1` or `ohs1` with `WEBHOST2` or `ohs2` in the instructions as necessary.

Creating the Virtual Host Configuration Files for Oracle Business Intelligence

You can route the Oracle HTTP Server requests to the Oracle Business Intelligence servers by creating the host configuration files.

> **Note:**
>
> Do not update the host entries in the virtual host configuration files mentioned in this topic if you have not planned to scale out the BI deployment to that host. For example, if you do not plan to extend the BI domain to HOST2 (BIHOST2 for WSM setup), do not specify "BIHOST2" in the directives while creating the virtual host configuration files.

> **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. Sign 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 WEBHOST1:7777>
   ServerName admin.example.com:80
   ServerAdmin you@your.address
   RewriteEngine On
   RewriteOptions inherit

# 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>

</VirtualHost>
```

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

#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 BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# BIEE Analytics
<Location /analytics>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

<Location /analytics-ws>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

<Location /bicontent>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

<Location /mobile>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

<Location /va>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# MapViewer
<Location /mapviewer>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# BI Publisher
<Location /xmlpserver>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# BI Search
<Location /bisearch>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# BI Composer
<Location /bicomposer>
   SetHandler weblogic-handler
```

```
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# EPM Provider Services
<Location /aps>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# EPM Workspace
<Location /workspace>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# BI SOA Services
<Location /biservices>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# AdminService
<Location /AdminService>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

# AsyncAdminService
<Location /AsyncAdminService>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

#OWSM
<Location /wsm-pm>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

#BI Security
<Location /bi-security>
    SetHandler weblogic-handler
    WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
</Location>

</VirtualHost>
```

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

```
<VirtualHost WEBHOST1:7777>
   ServerName https://bi.example.com:443
   ServerAdmin you@your.address
   RewriteEngine On
   RewriteOptions inherit
```

```
#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 BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /analytics-ws>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /bicontent>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /mobile>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

<Location /va>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# MapViewer
<Location /mapviewer>
```

```
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# BI Publisher
<Location /xmlpserver>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# BI Search
<Location /bisearch>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# BI Composer

<Location /bicomposer>
   SetHandler weblogic-handler
   WebLogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# EPM Provider Services
<Location /aps>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# EPM Workspace
<Location /workspace>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>

# OWSM
<Location /wsm-pm>
   SetHandler weblogic-handler
   WeblogicCluster BIHOST1VHN:7003,BIHOST2VHN:7003
   WLProxySSL ON
   WLProxySSLPassThrough ON
</Location>
```

```
</VirtualHost>
```

5. Restart the ohs1 instance:
   a. Change to the following directory:

      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 bi_vh.conf) to the configuration directory for the second Oracle HTTP Server instance (ohs2) on WEBHOST2:

   OHS_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf

7. Edit the .conf files and change any references from WEBHOST1 to WEBHOST2 in the <VirtualHost> directives.
8. Restart the ohs2 instance:
   a. Change to the following directory:

      OHS_DOMAIN_HOME/bin

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

      ```
      ./stopComponent.sh ohs2
      ./startComponent.sh ohs2
      ```

Configuring the WebLogic Proxy Plug-In

Before you can validate that requests that are routed correctly through the Oracle HTTP Server instances, you must set the WebLogic Plug-In Enabled parameter for the clusters that you just configured. To configure the WebLogic Proxy Plug-in:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Domain Structure** pane, expand the **Environment** node.
3. Click **Lock & Edit** in the Change Center.
4. Click **Clusters**.
5. Select the cluster to which you want to proxy requests from Oracle HTTP Server.

   The **Configuration: General** tab is displayed.
6. Scroll down to the **Advanced** section and expand it.
7. Set **WebLogic Plug-In Enabled** to **yes**.
8. Click **Save**.
9. If more than one cluster was deployed for the latest domain extension, repeat steps 4 through 8 until all the clusters are consistently updated.
10. Click **Activate Changes** in the Change Center.
11. Restart all Managed Servers in all the clusters that you modified in this chapter.

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`
- `https://bi.example.com/index.html`

Validating Access to the Management Consoles and Administration Server

To verify the changes that 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 by 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 by using a similar URL:

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

Validating 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:

- https://bi.example.com/analytics
- https://bi.example.com/mapviewer
- https://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

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully configure a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process. 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.

- 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.

Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers

If you have configured a separate shared storage volume or partition for secondary hosts, then you must install the Infrastructure on one of those hosts.

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

To install the software on the other host computers in the topology, log in to each host, and use the instructions in Starting the Infrastructure Installer 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 that the web servers are installed and configured in standalone mode, so they are not considered part of the application tier domain. See Configuring Oracle HTTP Server for an Enterprise Deployment

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.

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.

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.

- 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

Stopping the System Components

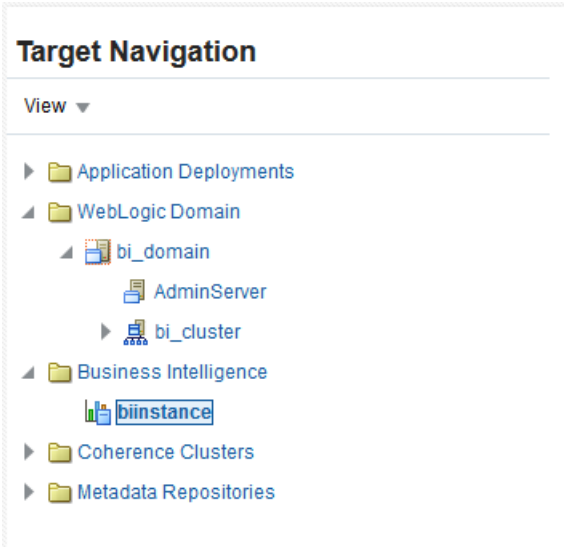
Follow these steps to stop the system components using Fusion Middleware Control.

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

   `http://ADMINVHN:7001/em`

2. Sign in to the 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**.

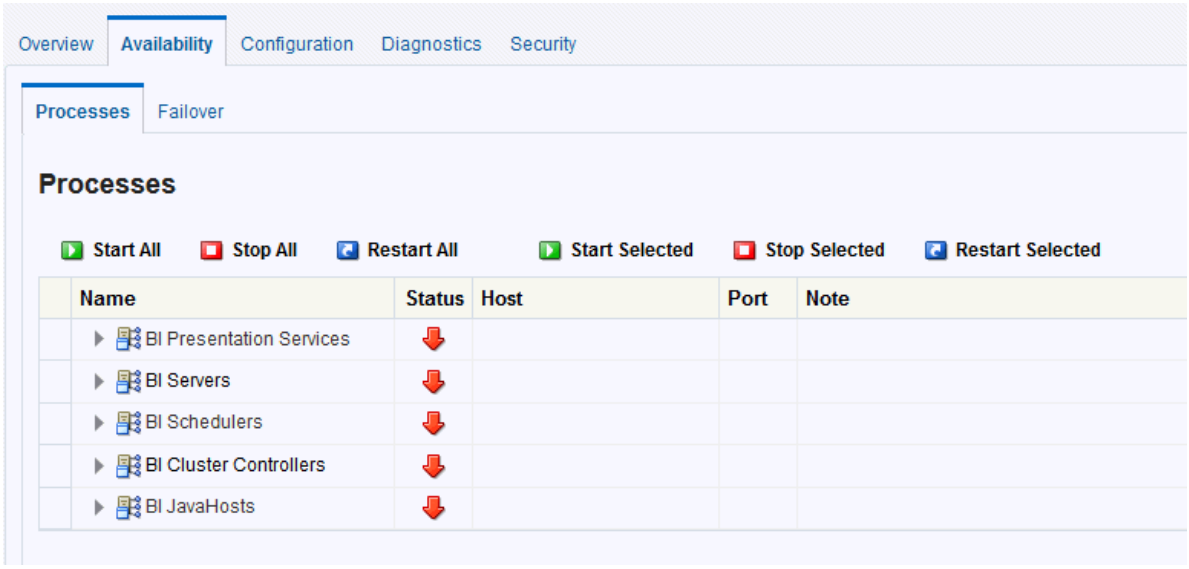
Figure 12-1 Selecting biinstance from the Target Navigation Pane



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.

Figure 12-2 Stopping All System Components



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:

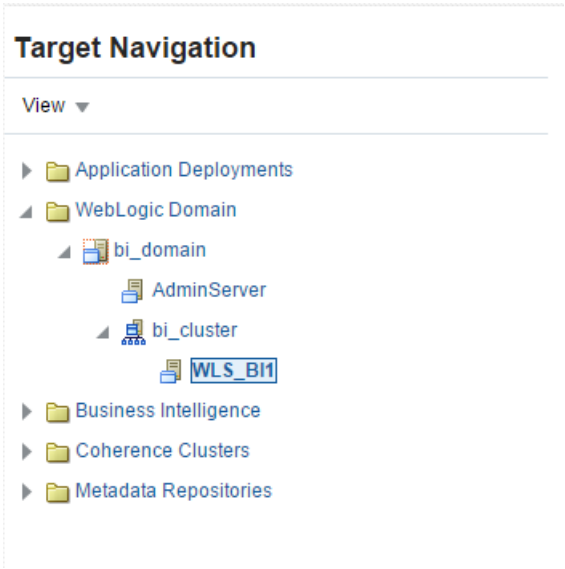
   `http://ADMINVHN:7001/em`

2. Sign in to the 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.

Figure 12-3 Selecting the Managed Server from the Target Navigation Pane



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.

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()
```

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
   ```

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
   ```

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','bihost2vhn.example.com',baseMachine='BIHOST1',base
   Server='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 `bihost2vhn.example.com` with the listen address of the new machine, BIHOST2VHN.
- `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()
```

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. Sign 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*.

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.

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.

Before starting the components on BIHOST1 and BIHOST2, complete the following tasks on BIHOST2:

- Disable the Derby database. See Disabling the Derby Database.
- Create a separate domain directory for managed servers. See Creating a Separate Domain Directory for Managed Servers on BIHOST1.
- Configure security for Essbase. See Configuring Security for Essbase in Oracle Business Intelligence.

.

To start the components after scale out, perform the following tasks.

- Starting the Node Manager in the Administration Server Domain Home
- Starting the Administration Server
- Starting the Node Managers in the Managed Server Domain Directories
- Starting the Managed Servers
- Starting the System Components

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.

Starting the Administration Server

To start the Administration Server using the Node Manager, use the procedure in Starting the Administration Server Using the Node Manager.

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. When following the steps in the Starting the Node Manager in the Managed Server Domain Directory on BIHOST1 section, substitute the value of BIHOST2 where BIHOST1 is used.

Starting the Managed Servers

Before starting the new WLS_BI2 Managed Server, verify that the Node Manager listen address for the machine associated with this new Managed Server is set to the value for BIHOST2. To do this, complete the following steps:

1. Sign in to Fusion Middleware Control using the Administration Server credentials.
2. Select the drop-down for **WebLogic Domain**. Select **Environment** and then **Machines**.
3. Select the machine BIHOST2.
4. Click the **Configuration** tab and then the **Node Manager** tab.
5. Click **Lock & Edit** in the **Change Center** menu at the top right corner of the page.
6. Change the listen address to be the value for BIHOST2.
7. Click **Save**.
8. Click **Activate Changes** in the **Change Center** menu at the top right corner of the page.

To start the WLS_BI1 and WLS_BI2 Managed Servers, use the procedure in Starting the WLS_BI1 Managed Server on BIHOST1.

Starting the System Components

To start all of the Oracle Business Intelligence system components, use the procedure in Starting the System Components.

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://BIHOST2VHN: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://BIHOST2VHN: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://BIHOST2VHN:7003/aps/Essbase`

Configuring Oracle BI Publisher

Perform these manual tasks to configure Oracle BI Publisher.

- 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).
- Configuring Integration with Oracle BI Presentation Services
- 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.
- Configuring BIPJmsResource for the Oracle BI Cluster
  To configure the BIPJMSResource JMS Module, you must modify some default values within the module as listed in this topic.

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. Sign in to BI Publisher using one of the following URLs:
   - http://BIHOST1VHN:7003/xmlpserver
   - http://BIHOST2VHN: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.

Configuring Integration with Oracle BI Presentation Services

To configure BI Publisher integration with Oracle BI Presentation Services:

1. Log into BI Publisher with Administrator credentials and select the **Administration** tab.
2. Under Integration, select **Oracle BI Presentation Services**.
3. Select **Manual Configuration** option. Then verify and update the following:
   - Server Protocol: `HTTP`
   - Server: `biinternal.mycompany.com`
   - Port: `80`
   - URL Suffix: `analytics-ws/saw.dll`
4. Click **Apply**.
5. Restart the BI Publisher application.

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. Sign in to BI Publisher using the following URL with administrator credentials.

   `http://BIHOST1VHN: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;
   ```

To find the cluster controller port number:

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

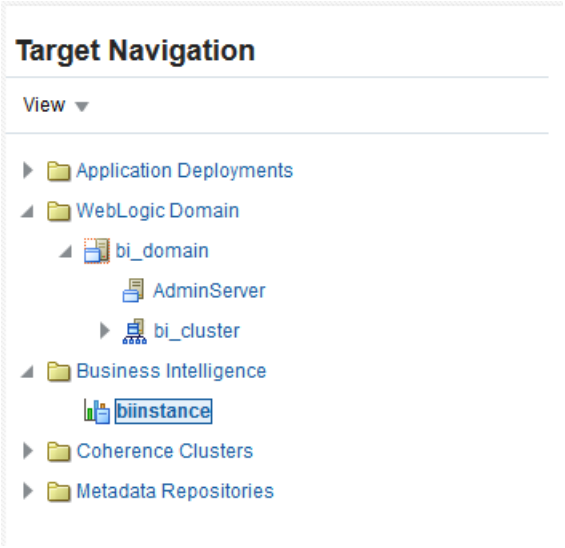
   `http://ADMINVHN:7001/em`

b. Sign in to the 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**.

Figure 12-4 Selecting biinstance from the Target Navigation Pane



The Business Intelligence Overview page appears.

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

f. Note the port number from the **Port** column.

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**.

Configuring BIPJmsResource for the Oracle BI Cluster

To configure the BIPJMSResource JMS Module, you must modify some default values within the module as listed in this topic.

The BIPJMSResource JMS Module is deployed automatically when you configure Oracle Business Intelligence Publisher in an Oracle WebLogic Server domain. However, you must modify the default values for the forwarding policy for the distributed topic in a cluster configuration.

Table 12-1 Specifying Custom Values for Setting Forwarding Policy

Property Name	Description
JMS Resource	BIP distributed topic in a cluster configuration - dist_BIP.System.T_auto
Property	Forwarding Policy
Description	A distributed BIP topic in a cluster installation is configured by default with the Forwarding Policy set to **Partitioned**.
Recommended Setting	Change the Forwarding Policy to **Replicated**.

To modify the BIPJMSResource Resource settings:

1. Sign in to the Oracle WebLogic Server Administration Console.
2. From **Services**, click **Messaging** and select **JMS Modules** from the left navigation pane.
3. Click **BIPJMSResource** on the list of JMS Modules.
4. Select **dist_BIP.System.T_auto** on the Summary of Resources table.
5. Click **General** tab.
6. From the **Forwarding Policy** menu, select **Replicated**.
7. Click **Save**.
8. Click **Activate Changes**.

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.

See Performing Backups and Recoveries for an Enterprise Deployment.

Part IV

Common Configuration and Management Procedures for an Enterprise Deployment

There are certain configuration and management procedures that are recommended for a typical enterprise deployment.

The following topics contain configuration and management procedures that are required for a typical enterprise deployment.

- Common Configuration and Management Tasks for an Enterprise Deployment
  The configuration and management tasks that may need to be performed on the enterprise deployment environment are detailed in this section.
- Using Whole Server Migration and Service Migration in an Enterprise Deployment
  The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. The following sections explain how these features can be used in an Oracle Fusion Middleware enterprise topology.
- Configuring Single Sign-On for an Enterprise Deployment
  You need to configure the Oracle HTTP Server WebGate in order to enable single sign-on with Oracle Access Manager.

13

Common Configuration and Management Tasks for an Enterprise Deployment

The configuration and management tasks that may need to be performed on the enterprise deployment environment are detailed in this section.

- Verifying Manual Failover of the Administration Server
  In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from BIHOST1 and BIHOST2 are detailed in the following sections.
- Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer
  It is important to understand how to enable SSL communication between the middle tier and the hardware load balancer.
- Performing Backups and Recoveries for an Enterprise Deployment
  It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle Business Intelligence enterprise deployment.
- Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment
  The persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence.
- About JDBC Persistent Stores for Web Services
  By default, web services use the WebLogic Server default persistent store for persistence. This store provides high-performance storage solution for web services.
- Performing Backups and Recoveries for an Enterprise Deployment
  It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle Business Intelligence enterprise deployment.
- Setting the Front End Host and Port for a WebLogic Cluster
  You must set the front-end HTTP host and port for the Oracle WebLogic Server cluster that hosts the Oracle SOA Suite servers. You can specify these values in the Configuration Wizard while you are specifying the properties of the domain. However, when you add a SOA Cluster as part of an Oracle Business Intelligence enterprise deployment, Oracle recommends that you perform this task after you verify the SOA Managed Servers.

Verifying Manual Failover of the Administration Server

In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from BIHOST1 and BIHOST2 are detailed in the following sections.

Assumptions:

- The Administration Server is configured to listen on ADMINVHN, and not on localhost or on any other host's address.

  For more information about the ADMINVHN virtual IP address, see Reserving the Required IP Addresses for an Enterprise Deployment.

- These procedures assume that the Administration Server domain home (*ASERVER_HOME*) has been mounted on both host computers. This ensures that the Administration Server domain configuration files and the persistent stores are saved on the shared storage device.
- The Administration Server is failed over from 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 a virtual sub-interface (for example, eth0:1), to be available on BIHOST1 or 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 Server failover procedure.

- 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
  If you have configured the web tier to access AdminServer, it is important to verify that you can access the Administration Server after you perform a manual failover of the Administration Server, by using the standard administration URLs.
- Configuring Roles for Administration of an Enterprise Deployment
  In order to manage each product effectively within a single enterprise deployment domain, you must understand which products require specific administration roles or groups, and how to add a product-specific administration role to the Enterprise Deployment Administration group.
- 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.

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. 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 to check the virtual IP address at its CIDR:

      ```
      ip addr show dev ethX
      ```

      Where, `X` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr show dev eth0
      ```

   b. Run the following command as root on BIHOST1 (where X:Y is the current interface used by ADMINVHN):

      ```
      ip addr del ADMINVHN/CIDR dev ethX:Y
      ```

      Where, `X:Y` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr del 100.200.140.206/24 dev eth0:1
      ```

   c. Run the following command as root on BIHOST2:

      ```
      ip addr add ADMINVHN/CIDR dev ethX label ethX:Y
      ```

      Where, `X:Y` is the current interface used by ADMINVHN.

      For example:

      ```
      ip addr add 100.200.140.206/24 dev eth0 label eth0:1
      ```

      > **Note:**
      >
      > Ensure that the CIDR representing the netmask and interface to be used to match the available network configuration in BIHOST2.
      >
      > The name of the network interface device may something other than eth*X*, especially on systems with redundant bonded interfaces.

4. Update the routing tables using `arping`, for example:

   ```
   arping -b -A -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`

Validating Access to the Administration Server on BIHOST2 Through Oracle HTTP Server

If you have configured the web tier to access AdminServer, it is important to verify that you can access the Administration Server after you perform a manual failover of the Administration Server, by using the standard administration URLs.

From the load balancer, access the following URLs to ensure that you can access the Administration Server when it is running on 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.

Configuring Roles for Administration of an Enterprise Deployment

In order to manage each product effectively within a single enterprise deployment domain, you must understand which products require specific administration roles or groups, and how to add a product-specific administration role to the Enterprise Deployment Administration group.

Each enterprise deployment consists of multiple products. Some of the products have specific administration users, roles, or groups that are used to control administration access to each product.

However, for an enterprise deployment, which consists of multiple products, you can use a single LDAP-based authorization provider and a single administration user and group to control access to all aspects of the deployment. See Creating a New LDAP Authenticator and Provisioning a New Enterprise Deployment Administrator User and Group.

To be sure that you can manage each product effectively within the single enterprise deployment domain, you must understand which products require specific administration roles or groups, you must know how to add any specific product administration roles to the single, common enterprise deployment administration group, and if necessary, you must know how to add the enterprise deployment administration user to any required product-specific administration groups.

For more information, see the following topics.

- Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group

Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group

For products with a product-specific administration group, use the following procedure to add the enterprise deployment administration user (`weblogic_bi` to the group. This allows you to manage the product by using the enterprise manager administrator user:

1. Create an **ldif** file called `product_admin_group.ldif` similar to the following:

```
dn: cn=product-specific_group_name, cn=groups, dc=us, dc=oracle, dc=com
displayname: product-specific_group_display_name
objectclass: top
objectclass: groupOfUniqueNames
objectclass: orclGroup
uniquemember: cn=weblogic_bi,cn=users,dc=us,dc=oracle,dc=com
cn: product-specific_group_name
description: Administrators Group for the Domain
```

   Replace *product-specific_group_display_name* with the display name for the group that appears in the management console for the LDAP server and in the Oracle WebLogic Server Administration Console.

2. Use the **ldif** file to add the enterprise deployment administrator user to the product-specific administration group.

   For Oracle Unified Directory:

```
OUD_INSTANCE_HOME/bin/ldapmodify -a
                                 -D "cn=Administrator"
                                 -X
                                 -p 1389
                                 -f product_admin_group.ldif
```

   For Oracle Internet Directory:

```
OID_ORACLE_HOME/bin/ldapadd -h oid.example.com
                            -p 389
                            -D cn="orcladmin"
                            -w <password>
                            -c
                            -v
                            -f product_admin_group.ldif
```

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 directory (ASERVER_HOME).

3. Migrate the ADMINVHN virtual IP address to the second host:

   a. Run the following command as root on BIHOST2 to check the virtual IP address at its CIDR:

```
ip addr show dev ethX
```

Where, X is the current interface used by ADMINVHN.

For example:

```
ip addr show dev eth0
```

b. Run the following command as root on BIHOST2 (where X:Y is the current interface used by ADMINVHN):

```
ip addr del ADMINVHN/CIDR dev ethX:Y
```

Where, X:Y is the current interface used by ADMINVHN.

For example:

```
ip addr del 100.200.140.206/24 dev eth0:1
```

c. Run the following command as root on BIHOST1:

```
ip addr add ADMINVHN/CIDR dev ethX label ethX:Y
```

Where, X:Y is the current interface used by ADMINVHN.

For example:

```
ip addr add 100.200.140.206/24 dev eth0 label eth0:1
```

> **Note:**
>
> Ensure that the CIDR representing the netmask and interface to be used to match the available network configuration in BIHOST1.

4. Update the routing tables using `arping`, for example:

```
arping -b -A -c 3 -I eht0 100.200.140.206
```

5. Start the Node Manager in the Administration Server domain home on BIHOST1.
6. Start the Administration Server on BIHOST1.
7. Test that you can access the Administration Server on BIHOST1 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
```

Enabling SSL Communication Between the Middle Tier and the Hardware Load Balancer

It is important to understand how to enable SSL communication between the middle tier and the hardware load balancer.

> **Note:**
>
> The following steps are applicable if the hardware load balancer is configured with SSL and the front-end address of the system has been secured accordingly.

- When is SSL Communication Between the Middle Tier and Load Balancer Necessary?
- Generating Self-Signed Certificates Using the utils.CertGen Utility
- Creating an Identity Keystore Using the utils.ImportPrivateKey Utility
- Creating a Trust Keystore Using the Keytool Utility
- Importing the Load Balancer Certificate into the Truststore
- Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts
- Configuring Node Manager to Use the Custom Keystores
- Configuring WebLogic Servers to Use the Custom Keystores

When is SSL Communication Between the Middle Tier and Load Balancer Necessary?

In an enterprise deployment, there are scenarios where the software running on the middle tier must access the front-end SSL address of the hardware load balancer. In these scenarios, an appropriate SSL handshake must take place between the load balancer and the invoking servers. This handshake is not possible unless the Administration Server and Managed Servers on the middle tier are started by using the appropriate SSL configuration.

Generating Self-Signed Certificates Using the utils.CertGen Utility

This section describes the procedure to create self-signed certificates on BIHOST1. Create certificates for every app-tier host by using the network name or alias of each host.

The directory where keystores and trust keystores are maintained must be on shared storage that is accessible from all nodes so that when the servers fail over (manually or with server migration), the appropriate certificates can be accessed from the failover node. Oracle recommends that you use central or shared stores for the certificates used for different purposes (for example, SSL set up for HTTP invocations). See the information on filesystem specifications for the KEYSTORE_HOME location provided in About the Recommended Directory Structure for an Enterprise Deployment.

For information on using trust CA certificates instead, see the information about configuring identity and trust in *Administering Security for Oracle WebLogic Server*.

About Passwords

The passwords used in this guide are used only as examples. Use secure passwords in a production environment. For example, use passwords that include both uppercase and lowercase characters as well as numbers.

To create self-signed certificates:

1. Temporarily, set up your environment by running the following script:

   ```
   . WL_HOME/server/bin/setWLSEnv.sh
   ```

   Note that there is a dot(.) and space( ) preceding the script name in order to source the shell script in the current shell.

2. Verify that the *CLASSPATH* environment variable is set:

   ```
   echo $CLASSPATH
   ```

3. Verify that the shared configuration directory folder has been created and mounted to shared storage correctly, as described in Preparing the File System for an Enterprise Deployment.

   For example, use the following command to verify that the shared configuration directory is available to each host:

   ```
   df -h | grep -B1 SHARED_CONFIG_DIR
   ```

   Replace *SHARED_CONFIG_DIR* with the actual path to your shared configuration directory.

   You can also do a listing of the directory to ensure that it is available to the host:

   ```
   ls -al SHARED_CONFIG_DIR
   ```

4. Create the keystore home folder structure if does not already exist.

   For example:

   ```
   cd SHARED_CONFIG_DIR
   mkdir keystores
   chown oracle:oinstall keystores
   chmod 750 keystores
   export KEYSTORE_HOME=SHARED_CONFIG_DIR/keystores
   ```

5. Change directory to the keystore home:

   ```
   cd KEYSTORE_HOME
   ```

6. Run the `utils.CertGen` tool to create the certificates for hostnames or aliases used by the managed servers and node managers, one per host.

> **Note:**
>
> You must run the `utils.CertGen` tool to create certificates for all the other hosts that run the Manager Servers.

Syntax:

```
java utils.CertGen key_passphrase cert_file_name key_file_name [export |
domestic] [hostname]
```

Examples:

```
java utils.CertGen password ADMINVHN.example.com_cert \
      ADMINVHN.example.com_key domestic ADMINVHN.example.com

java utils.CertGen password BIHOST1.example.com_cert \
      BIHOST1.example.com_key domestic BIHOST1.example.com

java utils.CertGen password BIHOST1VHN.example.com_cert \
       BIHOST1VHN.example.com_key domestic BIHOST1VHN.example.com
```

7. Repeat the above step for all the remaining hosts used in the system (for example, BIHOST2 and BIHOST2VHN).
8. For Dynamic clusters, in addition to `ADMINVHN` and one certificate for each host, a certificate matching a wildcard URL should also be generated.

   For example:

```
java utils.CertGen password WILDCARD.example.com_cert \
WILDCARD.example.com_key domestic \*.example.com
```

Creating an Identity Keystore Using the utils.ImportPrivateKey Utility

This section describes how to create an Identity Keystore on BIHOST1.example.com.

In previous sections you have created certificates and keys that reside on shared storage. In this section, the certificate and private keys created earlier for all hosts and ADMINVHN are imported into a new Identity Store. Make sure that you use a different alias for each of the certificate/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


java utils.ImportPrivateKey\
     -certfile KEYSTORE_HOME/BIHOST1VHN.example.com_cert.pem\
     -keyfile KEYSTORE_HOME/BIHOST1VHN.example.com_key.pem\
     -keyfilepass password\
     -keystore appIdentityKeyStore.jks\
     -storepass password\
     -alias BIHOST1VHN\
     -keypass password
```

2. Repeat the java importPrivateKey command for each of the remaining host-specific certificate and key pairs.For example, BIHOST2 and BIHOST2VHN

> **Note:**
>
> Make sure to use a unique alias for each certificate/key pair imported.

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 that you always change the default password, as follows:

   ```
   keytool -storepasswd -new NewPassword -keystore TrustKeyStore -storepass
   Original_Password
   ```

   For example:

   ```
   keytool -storepasswd -new password -keystore appTrustKeyStore.jks -storepass
   changeit
   ```

3. Import the CA certificate into the `appTrustKeyStore` by using the keytool utility.

   The CA certificate `CertGenCA.der` is used to sign all certificates generated by the `utils.CertGen` tool and is located at *`WL_HOME`*`/server/lib` directory.

   Use the following syntax to import the certificate:

   ```
   keytool -import -v -noprompt -trustcacerts -alias AliasName -file CAFileLocation
   -keystore KeyStoreLocation -storepass KeyStore_Password
   ```

   For example:

   ```
   keytool -import -v -noprompt -trustcacerts -alias clientCACert -file WL_HOME/
   server/lib/CertGenCA.der -keystore appTrustKeyStore.jks -storepass password
   ```

Importing the Load Balancer Certificate into the Truststore

For the SSL handshake to act properly, the load balancer's certificate must be added to the WLS servers truststore. To add a load balancer's certificate:

1. Access the site on SSL with a browser (this adds the server's certificate to the browser's repository).
2. Obtain the certificate from the load balancer. You can obtain the load balancer certificate using a browser such as Firefox. However, the easiest way to obtain the certificate is to use the `openssl` command. The syntax of the command is as follows:

```
openssl s_client -connect LOADBALANCER -showcerts </dev/null 2>/dev/
null|openssl x509 -outform PEM > SHARED_CONFIG_DIR/keystores/
LOADBALANCER.pem
```

For example:

```
openssl s_client -connect login.example.com:443 -showcerts </dev/null
2>/dev/null|openssl x509 -outform PEM > SHARED_CONFIG_DIR/keystores/
login.example.com.pem
```

3. Use the keytool to import the load balancer's certificate into the truststore:

   For example:

   ```
   keytool -import -file /oracle/certificates/bi.example.com.crt -v -keystore
   appTrustKeyStore.jks
   ```

> **Note:**
>
> The need to add the load balancer certificate to the WLS server truststore applies only to self-signed certificates. If the load balancer certificate is issued by a third-party CA, you have to import the public certificates of the root and the intermediate CAs into the truststore.

Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts

The `setDomainEnv.sh` script is provided by Oracle WebLogic Server and is used to start the Administration Server and the Managed Servers in the domain. To ensure that each server accesses 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 `DemoTrust.jks` entry with the following entry:

> **Note:**
>
> All the values for `EXTRA_JAVA_PROPERTIES` must be on one line in the file, followed by the export command on a new line.

```
EXTRA_JAVA_PROPERTIES="-Djavax.net.ssl.trustStore=/u01/oracle/config/keystores/
appTrustKeyStore.jks ${EXTRA_JAVA_PROPERTIES} ......."
export EXTRA_JAVA_PROPERTIES
```

3. Make the same change to the `setDomainEnv.sh` file in the *MSERVER_HOME*`/bin` directory BIHOST1, BIHOST2, `WEBHOST1, and WEBHOST2`.

> **Note:**
>
> The `setDomainEnv.sh` file cannot be copied between *ASERVER_HOME*`/bin` and *MSERVER_HOME*`/bin` as there are differences in the files for these two domain home locations. The*MSERVER_HOME*`/bin/setDomainEnv.sh` file can be copied between hosts.
>
> WebLogic Serverautomatically overwrites the `setDomainEnv.sh` file after each domain extension. Some patches may also replace this file. Verify your customizations to `setDomainEnv.sh` after each of these types of maintenance operations.

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, in the BIHOST1 *MSERVER_HOME*, use the alias BIHOST1 and in the *ASERVER_HOME* on BIHOST1, use the alias *ADMINVHN* according to the steps in Creating an Identity Keystore Using the utils.ImportPrivateKey Utility.

```
Example for BIHOST1:
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityKeyStoreFileName=KEYSTORE_HOME/appIdentityKeyStore.jks
CustomIdentityKeyStorePassPhrase=password
CustomIdentityAlias=BIHOST1
CustomIdentityPrivateKeyPassPhrase=password
```

The passphrase entries in the `nodemanager.properties` file are encrypted when you start Node Manager. For security reasons, minimize the time the entries in the `nodemanager.properties` file are left unencrypted. After you edit the file, restart Node Manager as soon as possible so that the entries are encrypted.

> **Note:**
>
> The `CustomIdentityAlias` value will need to be corrected every time the domain is extended after this configuration is performed. An unpack operation will replace the `CustomIdentityAlias` with the Administration Server's value when the domain configuration is written.

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 as the New_Password value in `Creating a Trust Keystore Using the Keytool Utility`.

     As mentioned in the previous step, this attribute may be optional or required depending on the type of keystore.
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 Administration Server.
16. Restart the Managed Servers where the keystore has been updated.

> **Note:**
>
> 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.

Performing Backups and Recoveries for an Enterprise Deployment

It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle Business Intelligence enterprise deployment.

> **Note:**
>
> Some of the static and runtime artifacts listed in this section are hosted from Network Attached Storage (NAS). If possible, backup and recover these volumes from the NAS filer directly rather than from the application servers.

For general information about backing up and recovering Oracle Fusion Middleware products, see the following sections in *Administering Oracle Fusion Middleware*:

- Backing Up Your Environment
- Recovering Your Environment

Table 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 (or NAS Filer)	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 (or NAS Filer)	Application Tier
Application home (APPLICATION_HOME)	BIHOST1 (or NAS Filer)	Application Tier
Oracle RAC databases	DBHOST1 and DBHOST2	Data Tier
Scripts and Customizations	Per host	Application Tier
Deployment Plan home (DEPLOY_PLAN_HOME)	BIHOST1 (or NAS Filer)	Application Tier
OHS/OTD Configuration directory	WEBHOST1 and WEBHOST2	Web Tier
Singleton Data Directory (SDD)	BIHOST1 (or NAS filer)	Application Tier

Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment

The persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence.

For example, the JMS subsystem stores persistent JMS messages and durable subscribers, and the JTA Transaction Log (TLOG) stores information about the committed transactions that are coordinated by the server but may not have been completed. The persistent store supports persistence to a file-based store or to a JDBC-enabled database. Persistent stores' high availability is provided by server or service migration. Server or service migration requires that all members of a WebLogic cluster have access to the same transaction and JMS persistent stores (regardless of whether the persistent store is file-based or database-based).

For an enterprise deployment, Oracle recommends using JDBC persistent stores for transaction logs (TLOGs) and JMS.

This section analyzes the benefits of using JDBC versus File persistent stores and explains the procedure for configuring the persistent stores in a supported database. If you want to use File persistent stores instead of JDBC stores, the procedure for configuring them is also explained in this section.

- Products and Components that use JMS Persistence Stores and TLOGs
- JDBC Persistent Stores vs. File Persistent Stores
- Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment
- Using File Persistent Stores for TLOGs and JMS in an Enterprise Deployment

Products and Components that use JMS Persistence Stores and TLOGs

Determining which installed FMW products and components utilize persistent stores can be done through the WebLogic Server Console in the Domain Structure navigation under ***DomainName*** > **Services** > **Persistent Stores**. The list indicates the name of the store, the store type (FileStore and JDBC), and the target of the store. The stores listed that pertain to MDS are outside the scope of this chapter and should not be considered.

These components (as applicable) use stores by default:

Component/Product	JMS Stores	TLOG Stores
B2B	Yes	Yes
BAM	Yes	Yes
BPM	Yes	Yes
ESS	No	No
HC	Yes	Yes
Insight	Yes	Yes
MFT	Yes	Yes
OSB	Yes	Yes
SOA	Yes	Yes
WSM	No	No

Component/Product	JMS Stores	TLOG Stores
OAM	No	No
OIM	Yes	Yes

JDBC Persistent Stores vs. File Persistent Stores

Oracle Fusion Middleware supports both database-based and file-based persistent stores for Oracle WebLogic Server transaction logs (TLOGs) and JMS. Before you decide on a persistent store strategy for your environment, consider the advantages and disadvantages of each approach.

> **Note:**
>
> Regardless of which storage method you choose, Oracle recommends that for transaction integrity and consistency, you use the same type of store for both JMS and TLOGs.

- About JDBC Persistent Stores for JMS and TLOGs
- Performance Considerations for TLOGs and JMS Persistent Stores

About JDBC Persistent Stores for JMS and TLOGs

When you store your TLOGs and JMS data in an Oracle database, you can take advantage of the replication and high availability features of the database. For example, you can use Oracle Data Guard to simplify cross-site synchronization. This is especially important if you are deploying Oracle Fusion Middleware in a disaster recovery configuration.

Storing TLOGs and JMS data in a database also means that you do not have to identity a specific shared storage location for this data. Note, however, that shared storage is still required for other aspects of an enterprise deployment. For example, it is necessary for Administration Server configuration (to support Administration Server failover), for deployment plans, and for adapter artifacts, such as the File and FTP Adapter control and processed files.

If you are storing TLOGs and JMS stores on a shared storage device, then you can protect this data by using the appropriate replication and backup strategy to guarantee zero data loss, and you potentially realize better system performance. However, the file system protection is always inferior to the protection provided by an Oracle Database.

For more information about the potential performance impact of using a database-based TLOGs and JMS store, see Performance Considerations for TLOGs and JMS Persistent Stores.

Performance Considerations for TLOGs and JMS Persistent Stores

One of the primary considerations when you select a storage method for Transaction Logs and JMS persistent stores is the potential impact on performance. This topic provides some guidelines and details to help you determine the performance impact of using JDBC persistent stores for TLOGs and JMS.

Performance Impact of Transaction Logs Versus JMS Stores

For transaction logs, the impact of using a JDBC store is relatively small, because the logs are very transient in nature. Typically, the effect is minimal when compared to other database operations in the system.

On the other hand, JMS database stores can have a higher impact on performance if the application is JMS intensive.

Factors that Affect Performance

There are multiple factors that can affect the performance of a system when it is using JMS DB stores for custom destinations. The main ones are:

- Custom destinations involved and their type
- Payloads being persisted
- Concurrency on the SOA system (producers on consumers for the destinations)

Depending on the effect of each one of the above, different settings can be configured in the following areas to improve performance:

- Type of data types used for the JMS table (using raw versus lobs)
- Segment definition for the JMS table (partitions at index and table level)

Impact of JMS Topics

If your system uses Topics intensively, then as concurrency increases, the performance degradation with an Oracle RAC database will increase more than for Queues. In tests conducted by Oracle with JMS, the average performance degradation for different payload sizes and different concurrency was less than 30% for Queues. For topics, the impact was more than 40%. Consider the importance of these destinations from the recovery perspective when deciding whether to use database stores.

Impact of Data Type and Payload Size

When you choose to use the RAW or SecureFiles LOB data type for the payloads, consider the size of the payload being persisted. For example, when payload sizes range between 100b and 20k, then the amount of database time required by SecureFiles LOB is slightly higher than for the RAW data type.

More specifically, when the payload size reach around 4k, then SecureFiles tend to require more database time. This is because 4k is where writes move out-of-row. At around 20k payload size, SecureFiles data starts being more efficient. When payload sizes increase to more than 20k, then the database time becomes worse for payloads set to the RAW data type.

One additional advantage for SecureFiles is that the database time incurred stabilizes with payload increases starting at 500k. In other words, at that point it is not relevant (for SecureFiles) whether the data is storing 500k, 1MB or 2MB payloads, because the write is asynchronized, and the contention is the same in all cases.

The effect of concurrency (producers and consumers) on the queue's throughput is similar for both RAW and SecureFiles until the payload sizes reach 50K. For small payloads, the effect on varying concurrency is practically the same, with slightly better scalability for RAW. Scalability is better for SecureFiles when the payloads are above 50k.

Impact of Concurrency, Worker Threads, and Database Partioning

Concurrency and worker threads defined for the persistent store can cause contention in the RAC database at the index and global cache level. Using a reverse index when enabling multiple worker threads in one single server or using multiple Oracle WebLogic Server clusters can improve things. However, if the Oracle Database partitioning option is available, then global hash partition for indexes should be used instead. This reduces the contention on the index and the global cache buffer waits, which in turn improves the response time of the application. Partitioning works well in all cases, some of which will not see significant improvements with a reverse index.

Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment

This section explains the guidelines to use JDBC persistent stores for transaction logs (TLOGs) and JMS. It also explains the procedures to configure the persistent stores in a supported database.

- Recommendations for TLOGs and JMS Datasource Consolidation
  To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.
- Roadmap for Configuring a JDBC Persistent Store for TLOGs
  The following topics describe how to configure a database-based persistent store for transaction logs.
- Roadmap for Configuring a JDBC Persistent Store for JMS
  The following topics describe how to configure a database-based persistent store for JMS.
- Creating a User and Tablespace for TLOGs
  Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.
- Creating a User and Tablespace for JMS
  Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.
- Creating GridLink Data Sources for TLOGs and JMS Stores
  Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.
- Assigning the TLOGs JDBC Store to the Managed Servers
  After you create the tablespace and user in the database, and you have created the datasource, you can then assign the TLOGs persistence store to each of the required Managed Servers.
- Creating a JDBC JMS Store
  After you create the JMS persistent store user and table space in the database, and after you create the data source for the JMS persistent store, you can then use the Administration Console to create the store.
- Assigning the JMS JDBC store to the JMS Servers
  After you create the JMS tablespace and user in the database, create the JMS datasource, and create the JDBC store, then you can assign the JMS persistence store to each of the required JMS Servers.
- Creating the Required Tables for the JMS JDBC Store
  The final step in using a JDBC persistent store for JMS is to create the required JDBC store tables. Perform this task before you restart the Managed Servers in the domain.

Recommendations for TLOGs and JMS Datasource Consolidation

To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.

Oracle recommends you to reuse the `WLSSchemaDatasource` as is for TLOGs and JMS persistent stores under non-high workloads and consider increasing the `WLSSchemaDatasource` pool size. Reuse of datasource forces to use the same schema and tablespaces, and so the `PREFIX_WLS_RUNTIME` schema in the `PREFIX_WLS` tablespace is used for both TLOGs and JMS messages.

High stress (related with high JMS activity, for example) and contention in the datasource can cause stability and performance problems. For example:

- High contention in the DataSource can cause persistent stores to fail if no connections are available in the pool to persist JMS messages.
- High Contention in the DataSource can cause issues in transactions if no connections are available in the pool to update transaction logs.

For these cases, use a separate datasource for TLOGs and stores and a separate datasource for the different stores. You can still reuse the `PREFIX_WLS_RUNTIME` schema but configure separate custom datasources to the same schema to solve the contention issue.

Roadmap for Configuring a JDBC Persistent Store for TLOGs

The following topics describe how to configure a database-based persistent store for transaction logs.

1. Creating a User and Tablespace for TLOGs
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Assigning the TLOGs JDBC Store to the Managed Servers

> **Note:**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Roadmap for Configuring a JDBC Persistent Store for JMS

The following topics describe how to configure a database-based persistent store for JMS.

1. Creating a User and Tablespace for JMS
2. Creating GridLink Data Sources for TLOGs and JMS Stores
3. Creating a JDBC JMS Store
4. Assigning the JMS JDBC store to the JMS Servers
5. Creating the Required Tables for the JMS JDBC Store

> **Note:**
>
> Steps 1 and 2 are optional. To accomplish data source consolidation and connection usage reduction, you can reuse `PREFIX_WLS` tablespace and `WLSSchemaDatasource` as described in Recommendations for TLOGs and JMS Datasource Consolidation.

Creating a User and Tablespace for TLOGs

Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.

1. Create a tablespace called `tlogs`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace tlogs
           logging datafile 'path-to-data-file-or-+asmvolume'
           size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `TLOGS` and assign to it the `tlogs` tablespace.

   For example:

   ```
   SQL> create user TLOGS identified by password;

   SQL> grant create table to TLOGS;

   SQL> grant create session to TLOGS;

   SQL> alter user TLOGS default tablespace tlogs;

   SQL> alter user TLOGS quota unlimited on tlogs;
   ```

Creating a User and Tablespace for JMS

Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.

1. Create a tablespace called `jms`.

   For example, log in to SQL*Plus as the `sysdba` user and run the following command:

   ```
   SQL> create tablespace jms
           logging datafile 'path-to-data-file-or-+asmvolume'
           size 32m autoextend on next 32m maxsize 2048m extent management local;
   ```

2. Create a user named `JMS` and assign to it the `jms` tablespace.

   For example:

   ```
   SQL> create user JMS identified by password;

   SQL> grant create table to JMS;
   ```

```
SQL> grant create session to JMS;

SQL> alter user JMS default tablespace jms;

SQL> alter user JMS quota unlimited on jms;
```

Creating GridLink Data Sources for TLOGs and JMS Stores

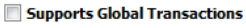
Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.

For an enterprise deployment, you should use GridLink data sources for your TLOGs and JMS stores. To create a GridLink data source:

1. Sign in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Data Sources**.
4. On the Summary of Data Sources page, click **New** and select **GridLink Data Source**, and enter the following:
   - Enter a logical name for the data source in the **Name** field.

     For the TLOGs store, enter TLOG; for the JMS store, enter JMS.
   - Enter a name for **JNDI**.

     For the TLOGs store, enter `jdbc/tlogs`; for the JMS store, enter `jdbc/jms`.
   - For the Database Driver, select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
   - Click **Next**.
5. In the Transaction Options page, clear the **Supports Global Transactions** check box, and then click **Next**.



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.

Figure 13-1 Adding Host Name and Port Details for the RAC Database

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

```
SQL>show parameter remote_listener;

NAME                 TYPE        VALUE

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

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

> **Note:**
>
> For Oracle Database 11*g* Release 1 (11.1), use the virtual IP and port of each database instance listener, for example:
>
> `dbhost1-vip.example.com (port 1521)`
>
> and
>
> `dbhost2-vip.example.com (1521)`

- **Database User Name**: Enter the following:

  For the TLOGs store, enter `TLOGS`; for the JMS persistent store, enter `JMS`.
- **Password**: Enter the password that you used when you created the user in the database.
- **Confirm Password**: Enter the password again and click **Next**.

8. On the Test GridLink Database Connection page, review the connection parameters and click **Test All Listeners**.

   Here is an example of a successful connection notification:

   ```
   Connection test for
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=db-
   scan.example.com)
   (PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=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: ONS remote port for the RAC database and the ONS remote port as reported by the database (see the following example) and click **Add**:

```
[orcl@db-scan1 ~]$ srvctl config nodeapps -s

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

- Click **Next**.

> **Note:**
>
> For Oracle Database 11*g* Release 1 (11.1), use the hostname and port of each database's ONS service, for example:
>
> ```
> custdbhost1.example.com (port 6200)
> ```
>
> and
>
> ```
> custdbhost2.example.com (6200)
> ```

10. On the Test ONS Client Configuration page, review the connection parameters and click **Test All ONS Nodes**.

    Here is an example of a successful connection notification:

    ```
    Connection test for db-scan.example.com:6200 succeeded.
    ```

    Click **Next**.
11. In the Select Targets page, select the cluster that is using the persistent store, and then select **All Servers in the cluster**.
12. Click **Finish**.
13. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.
14. Repeat step 4 through step 13 to create the GridLink Data Source for JMS File Stores.

Assigning the TLOGs JDBC Store to the Managed Servers

After you create the tablespace and user in the database, and you have created the datasource, you can then assign the TLOGs persistence store to each of the required Managed Servers.

1. Login in to the Oracle WebLogic Server Administration Console.
2. In the **Change Center**, click **Lock and Edit**.
3. In the Domain Structure tree, expand **Environment**, then **Servers**.
4. Click the name of the Managed Server you want to use the TLOGs store.
5. Select the **Configuration** > **Services** tab.
6. Under **Transaction Log Store**, select **JDBC** from the **Type** menu.
7. From the **Data Source** menu, select the data source you created for the TLOGs persistence store.

8. In the **Prefix Name** field, specify a prefix name to form a unique JDBC TLOG store name for each configured JDBC TLOG store
9. Click **Save**.
10. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Creating a JDBC JMS Store

After you create the JMS persistent store user and table space in the database, and after you create the data source for the JMS persistent store, you can then use the Administration Console to create the store.

1. Log in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Persistent Store**.
4. Click **New**, and then click **JDBC Store**.
5. Enter a persistent store name that easily relates it to the pertaining JMS servers that will be using it.
6. Specify a unique prefix qualifying the installation and cluster and associate it with the data source you created for the JMS persistent store.
7. Target the store to the entity that will host the JTA services.
8. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Assigning the JMS JDBC store to the JMS Servers

After you create the JMS tablespace and user in the database, create the JMS datasource, and create the JDBC store, then you can assign the JMS persistence store to each of the required JMS Servers.

To assign the JMS persistence store to the JMS servers:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the **Change Center**, click **Lock and Edit**.
3. In the Domain Structure tree, expand **Services**, then **Messaging**, and then **JMS Servers**.
4. Click the name of the JMS Server that you want to use the persistent store.
5. From the **Persistent Store** menu, select the JMS persistent store you created earlier.
6. Click **Save**.
7. Repeat steps 3 to 6 for each of the additional JMS Servers in the cluster.
8. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

Creating the Required Tables for the JMS JDBC Store

The final step in using a JDBC persistent store for JMS is to create the required JDBC store tables. Perform this task before you restart the Managed Servers in the domain.

1. Review the information in Performance Considerations for TLOGs and JMS Persistent Stores, and decide which table features are appropriate for your environment.

   There are three Oracle DB schema definitions provided in this release and were extracted for review in the previous step. The basic definition includes the RAW data type without any partition for indexes. The second uses the blob data type, and the third uses the blob data type and secure files.

2. Create a domain-specific well-named folder structure for the custom `DDL` file on shared storage. The *ORACLE_RUNTIME* shared volume is recommended so it is available to all servers.

   Example:

   ```
   mkdir -p ORACLE_RUNTIME/domain_name/ddl
   ```

3. Create a `jms_custom.ddl` file in new shared `ddl` folder based on your requirements analysis.

   For example, to implement an optimized schema definition that uses both secure files and hash partitioning, create the `jms_custom.ddl` file with the following content:

   ```
   CREATE TABLE $TABLE (
     id     int  not null,
     type   int  not null,
     handle int  not null,
     record blob not null,
   PRIMARY KEY (ID) USING INDEX GLOBAL PARTITION BY HASH (ID) PARTITIONS 8)
   LOB (RECORD) STORE AS SECUREFILE (ENABLE STORAGE IN ROW);
   ```

   This example can be compared to the default schema definition for JMS stores, where the RAW data type is used without any partitions for indexes.

   Note that the number of partitions should be a power of two. This ensures that each partition is of similar size. The recommended number of partitions varies depending on the expected table or index growth. You should have your database administrator (DBA) analyze the growth of the tables over time and adjust the tables accordingly. See Partitioning Concepts in *Database VLDB and Partitioning Guide*.

4. Use the Administration Console to edit the existing JDBC Store you created earlier; create the table that is used for the JMS data:

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

   b. In the **Change Center**, click **Lock and Edit**.

   c. In the Domain Structure tree, expand **Services**, then **Persistent Stores**.

   d. Click the persistent store you created earlier.

e. Under the **Advanced** options, enter *ORACLE_RUNTIME*/*domain_name*/ddl/jms_custom.ddl in the **Create Table from DDL File** field.

f. Click **Save**.

g. To activate these changes, in the Change Center of the Administration Console, click **Activate Changes**.

5. Restart the Managed Servers.

Using File Persistent Stores for TLOGs and JMS in an Enterprise Deployment

This section explains the procedures to configure TLOGs and JMS File persistent stores in a shared folder.

- Configuring TLOGs File Persistent Store in a Shared Folder
- Configuring JMS File Persistent Store in a Shared Folder

Configuring TLOGs File Persistent Store in a Shared Folder

Oracle WebLogic Server uses the transaction logs to recover from system crashes or network failures.

- Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster
- Validating the Location and Creation of the Transaction Logs

Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster

To set the location for the default persistence stores for each managed server in a static cluster, complete the following steps:

1. Log into the Oracle WebLogic Server Administration console:

   ADMINVHN:7001/console

   > **Note:**
   >
   > If you have already configured web tier, use http://admin.example.com/console.

2. In the Change Center section, click **Lock & Edit**.
3. For each of the Managed Servers in the cluster:

   a. In the Domain Structure window, expand the **Environment** node, and then click the **Servers** node.

      The Summary of Servers page appears.

   b. Click the name of the server (represented as a hyperlink) in the **Name** column of the table.

      The settings page for the selected server appears and defaults to the Configuration tab.

   c. On the **Configuration** tab, click the **Services** tab.

d. In the Default Store section of the page, enter the path to the folder where the default persistent stores stores its data files.

   For the enterprise deployment, use the *ORACLE_RUNTIME* directory location. This subdirectory serves as the central, shared location for transaction logs for the cluster. See File System and Directory Variables Used in This Guide.

   For example:

   ```
   ORACLE_RUNTIME/domain_name/cluster_name/tlogs
   ```

   In this example, replace *ORACLE_RUNTIME* with the value of the variable for your environment. Replace *domain_name* with the name you assigned to the domain. Replace *cluster_name* with the name of the cluster you just created.

e. Click **Save**.

4. Complete step 3 for all servers in the SOA_Cluster.
5. Click **Activate Changes**.

> **Note:**
>
> You validate the location and the creation of the transaction logs later in the configuration procedure.

Validating the Location and Creation of the Transaction Logs

After the WLS_SERVER_TYPE1 and WLS_SERVER_TYPE2 Managed Servers are up and running, based on the steps that you performed in Configuring TLOGs File Persistent Store in a Shared Folder with a Static Cluster, verify that the following transaction log directory and transaction logs are created as expected:

```
ORACLE_RUNTIME/domain_name/OSB_Cluster/tlogs
```

- `_WLS_WLS_SERVER_TYPE1000000.DAT`
- `_WLS_WLS_SERVER_TYPE2000000.DAT`

Configuring JMS File Persistent Store in a Shared Folder

If you have already configured and extended your domain, the JMS Persistent Files are already configured in a shared location. If you need to change any other persistent store file to the shared folder, perform the following steps:

1. Log in to the Oracle WebLogic Server Administration Console.
2. Navigate to **Domain > Services > Persistent Store** and click the name of the persistent store that you want to move to the shared folder.

   The **Configuration: General** tab is displayed.
3. Change the directory to `ORACLE_RUNTIME/domain_name/bi_cluster/jms`.
4. Click **Save**.
5. Click **Activate Changes**.

About JDBC Persistent Stores for Web Services

By default, web services use the WebLogic Server default persistent store for persistence. This store provides high-performance storage solution for web services.

The default web service persistence store is used by the following advanced features:

- Reliable Messaging
- Make Connection
- SecureConversation
- Message buffering

You also have the option to use a JDBC persistence store in your WebLogic Server web service, instead of the default store. For information about web service persistence, see Managing Web Service Persistence.

Performing Backups and Recoveries for an Enterprise Deployment

It is recommended that you follow the below mentioned guidelines to make sure that you back up the necessary directories and configuration data for an Oracle Business Intelligence enterprise deployment.

> **Note:**
>
> Some of the static and runtime artifacts listed in this section are hosted from Network Attached Storage (NAS). If possible, backup and recover these volumes from the NAS filer directly rather than from the application servers.

For general information about backing up and recovering Oracle Fusion Middleware products, see the following sections in *Administering Oracle Fusion Middleware*:

- Backing Up Your Environment
- Recovering Your Environment

Table 13-1 lists the static artifacts to back up in a typical Oracle Business Intelligence enterprise deployment.

Table 13-3 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 (or NAS Filer)	Application Tier

Table 13-3 (Cont.) Static Artifacts to Back Up in the Oracle Business Intelligence Enterprise Deployment

Type	Host	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 13-4 Run-Time Artifacts to Back Up in the Oracle Business Intelligence Enterprise Deployment

Type	Host	Tier
Administration Server domain home (ASERVER_HOME)	BIHOST1 (or NAS Filer)	Application Tier
Application home (APPLICATION_HOME)	BIHOST1 (or NAS Filer)	Application Tier
Oracle RAC databases	DBHOST1 and DBHOST2	Data Tier
Scripts and Customizations	Per host	Application Tier
Deployment Plan home (DEPLOY_PLAN_HOME)	BIHOST1 (or NAS Filer)	Application Tier
OHS/OTD Configuration directory	WEBHOST1 and WEBHOST2	Web Tier
Singleton Data Directory (SDD)	BIHOST1 (or NAS filer)	Application Tier

Setting the Front End Host and Port for a WebLogic Cluster

You must set the front-end HTTP host and port for the Oracle WebLogic Server cluster that hosts the Oracle SOA Suite servers. You can specify these values in the Configuration Wizard while you are specifying the properties of the domain. However, when you add a SOA Cluster as part of an Oracle Business Intelligence enterprise deployment, Oracle recommends that you perform this task after you verify the SOA Managed Servers.

To set the frontend host and port from the Weblogic Server Administration Console:

1. Log in to the WebLogic Server Administration Console.
2. In the Change Center, click **Lock & Edit**.
3. In the Domain Structure panel, expand **Environment**, and click **Clusters**.
4. On the Clusters page, click the cluster that you want to modify, and then select the **HTTP** tab.
5. Set the following values:
   - **Cluster Name**: BI_Cluster
   - **Frontend Host**: `bi.example.com`
   - **Frontend HTTP Port**: `80`
   - **Frontend HTTPS Port**: `442`

6. Click **Save**.
7. Click **Activate Changes**.
8. Restart the managed servers of the cluster.

14

Using Whole Server Migration and Service Migration in an Enterprise Deployment

The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. The following sections explain how these features can be used in an Oracle Fusion Middleware enterprise topology.

- About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment
  Oracle WebLogic Server provides a migration framework that is an integral part of any highly available environment. The following sections provide more information about how this framework can be used effectively in an enterprise deployment.
- Converting to Virtual IP Addresses in Preparation for Whole Server Migration
  This section explains the steps necessary to configure each server, each machine, and each server's Node Manager to listen on the appropriate Virtual IP Address.
- Creating a GridLink Data Source for Leasing
  Whole Server Migration and Automatic Service Migration require a data source for the leasing table, which is a tablespace created automatically as part of the Oracle WebLogic Server schemas by the Repository Creation Utility (RCU).
- Configuring Whole Server Migration for an Enterprise Deployment
  After you have prepared your domain for whole server migration or automatic service migration, you can configure Whole Server Migration for specific Managed Servers within a cluster.

About Whole Server Migration and Automatic Service Migration in an Enterprise Deployment

Oracle WebLogic Server provides a migration framework that is an integral part of any highly available environment. The following sections provide more information about how this framework can be used effectively in an enterprise deployment.

- Understanding the Difference between Whole Server and Service Migration
- Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment
- Understanding Which Products and Components Require Whole Server Migration and Service Migration

Understanding the Difference between Whole Server and Service Migration

The Oracle WebLogic Server migration framework supports two distinct types of automatic migration:

- **Whole Server Migration**, where the Managed Server instance is migrated to a different physical system upon failure.

  Whole server migration provides for the automatic restart of a server instance, with all its services, on a different physical machine. When a failure occurs in a server that is part of a cluster which is configured with server migration, the server is restarted on any of the other machines that host members of the cluster.

  For this to happen, the servers must use a floating IP as listen address and the required resources (transactions logs and JMS persistent stores) must be available on the candidate machines.

  See Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.

- **Service Migration**, where specific services are moved to a different Managed Server within the cluster.

  To understand service migration, it's important to understand *pinned services*.

  In a WebLogic Server cluster, most subsystem services are hosted homogeneously on all server instances in the cluster, enabling transparent failover from one server to another. In contrast, pinned services, such as JMS-related services, the JTA Transaction Recovery Service, and user-defined singleton services, are hosted on individual server instances within a cluster—for these services, the WebLogic Server migration framework supports failure recovery with service migration, as opposed to failover.

  See Understanding the Service Migration Framework in *Administering Clusters for Oracle WebLogic Server*.

Implications of Using Whole Server Migration or Service Migration in an Enterprise Deployment

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.

Understanding Which Products and Components Require Whole Server Migration and Service Migration

Note that the table lists the recommended best practice. It does not preclude you from using Whole Server or Automatic Server Migration for those components that support it.

Component	Whole Server Migration (WSM)	Automatic Service Migration (ASM)
Oracle Business Intelligence Publisher	YES	NO

Converting to Virtual IP Addresses in Preparation for Whole Server Migration

This section explains the steps necessary to configure each server, each machine, and each server's Node Manager to listen on the appropriate Virtual IP Address.

If you have configured the BI Servers using the values for BIHOST1 and BIHOST2 (the physical hostnames), then you need to modify them to listen on a Virtual IP Address (BIHOST1VHN and BIHOST2VHN) as mentioned in Table 5-1.

Enabling the Required Virtual IP Addresses on Each Host

Follow the steps outlined in Enabling the Required Virtual IP Addresses on Each Host to enable the virtual IP addresses (BIHOST1VHN and BIHOST2VHN) for each BI WLS server.

Editing the Listen Address for the Managed Servers

To edit the listen address:

1. Sign in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure window, expand **Environment** and select **Servers**. The Summary of Servers page appears.
3. Select the server for which you want to configure migration in the Name column of the table, which is WLS_BI1.
4. Click **Lock & Edit**.
5. On the Configuration tab, under the General tab, modify the Listen Address to be the value of the VIP assigned to the WLS_BI1 server (BIHOST1VHN).
6. Click **Save**. Repeat the steps for each of the WLS Servers that you want to configure for migration.
7. Click **Activate Changes**.
8. Restart the servers.

Editing the OHS Virtual Host Configuration Files

To modify the OHS virtual host configuration files in order to reflect the change from physical hostname to virtual IPs:

1. Change to the following directory on each OHS server:

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

2. Edit the `biinternal_vh.conf` file and change the references of BIHOST1 to the value for BIHOST1VHN and the references of BIHOST1 to the value for BIHOST2VHN.
3. Edit the `bi_vh.conf` file and change the references of BIHOST1 to the value for BIHOST1VHN and the references of BIHOST1 to the value for BIHOST2VHN.
4. Restart the OHS servers.

Creating a GridLink Data Source for Leasing

Whole Server Migration and Automatic Service Migration require a data source for the leasing table, which is a tablespace created automatically as part of the Oracle WebLogic Server schemas by the Repository Creation Utility (RCU).

> **Note:**
>
> To accomplish data source consolidation and connection usage reduction, you can reuse the `WLSSchemaDatasource` as is for database leasing. This datasource is already configured with the `FMW1221_WLS_RUNTIME` schema, where the leasing table is stored.

For an enterprise deployment, you should create a GridLink data source:

1. Log in to the Oracle WebLogic Server Administration Console.
2. If you have not already done so, in the **Change Center**, click **Lock & Edit**.
3. In the **Domain Structure** tree, expand **Services**, then select **Data Sources**.
4. On the Summary of Data Sources page, click **New** and select **GridLink Data Source**, and enter the following:
   - Enter a logical name for the data source in the **Name** field. For example, **Leasing**.
   - Enter a name for **JNDI**. For example, **jdbc/leasing**.
   - For the Database Driver, select **Oracle's Driver (Thin) for GridLink Connections Versions: Any**.
   - Click **Next**.
5. In the Transaction Options page, clear the **Supports Global Transactions** check box, and then click **Next**.

6. In the GridLink Data Source Connection Properties Options screen, select **Enter individual listener information** and click **Next**.

7. Enter the following connection properties:

   - **Service Name**: Enter the service name of the database with lowercase characters. For a GridLink data source, you must enter the Oracle RAC service name. For example:

     ```
     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.

Figure 14-1 Specifying SCAN Address for the RAC Database

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**.

     This value should be the ONS host and ONS remote port for the RAC database. To find the ONS remote port for the database, you can use the following command on the database host:

     ```
     [orcl@db-scan1 ~]$ srvctl config nodeapps -s

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

   - Click **Next**.

> **Note:**
>
> For Oracle Database 11*g* Release 1 (11.1), use the hostname and port of each database's ONS service, for example:
>
> ```
> custdbhost1.example.com (port 6200)
> ```
>
> and
>
> ```
> custdbhost2.example.com (6200)
> ```

10. On the Test ONS Client Configuration page, review the connection parameters and click **Test All ONS Nodes**.

    Here is an example of a successful connection notification:

    ```
    Connection test for db-scan.example.com:6200 succeeded.
    ```

    Click **Next**.

11. In the Select Targets page, select the cluster that you are configuring for Whole Server Migration or Automatic Service Migration, and then select **All Servers in the cluster**.
12. Click **Finish**.
13. Click **Activate Changes**.

Configuring Whole Server Migration for an Enterprise Deployment

After you have prepared your domain for whole server migration or automatic service migration, you can configure Whole Server Migration for specific Managed Servers within a cluster.

> **Note:**
>
> As mentioned earlier, for migration to work, servers must use a virtual hostname that matches a floating IP, as the listen address. You can specify the listen address directly in the Configuration Wizard or update it in the administration console.

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

Editing the Node Manager's Properties File to Enable Whole Server Migration

Use the section to edit the Node Manager properties file on the two nodes where the servers are running.

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

   MSERVER_HOME`/nodemanager/nodmeanager.properties`
2. If not done already, set the `StartScriptEnabled` property in the `nodemanager.properties` file to true.

   This is required to enable Node Manager to start the managed servers.
3. Add the following properties to the `nodemanager.properties` file to enable server migration to work properly:
   - `Interface`

     `Interface=eth0`

     This property specifies the interface name for the floating IP (`eth0`, for example).

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

- `NetMask`

  ```
  NetMask=255.255.255.0
  ```

  This property specifies the net mask for the interface for the floating IP.

- `UseMACBroadcast`

  ```
  UseMACBroadcast=true
  ```

  This property specifies whether to use a node's MAC address when sending ARP packets, that is, whether to use the `-b` flag in the arping command.

4. Restart the Node Manager.
5. Verify in the output of Node Manager (the shell where the Node Manager is started) that these properties are in use. Otherwise, problems may occur during migration. The output should be similar to the following:

   ```
   ...
   SecureListener=true
   LogCount=1
   eth0=*,NetMask=255.255.255.0
   ...
   ```

Setting Environment and Superuser Privileges for the wlsifconfig.sh Script

Use this section to set the environment and superuser privileges for the `wlsifconfig.sh` script, which is used to transfer IP addresses from one machine to another during migration. It must be able to run `ifconfig`, which is generally only available to superusers.

For more information about the `wlsifconfig.sh` script, see Configuring Automatic Whole Server Migration in *Administering Clusters for Oracle WebLogic Server*.

Refer to the following sections for instructions on preparing your system to run the `wlsifconfig.sh` script.

- Setting the PATH Environment Variable for the wlsifconfig.sh Script
- Granting Privileges to the wlsifconfig.sh Script

Setting the PATH Environment Variable for the wlsifconfig.sh Script

Ensure that the commands listed in the following table are included in the PATH environment variable for each host computers.

File	Directory Location
`wlsifconfig.sh`	*MSERVER_HOME*`/bin/server_migration`
`wlscontrol.sh`	*WL_HOME*`/common/bin`
`nodemanager.domains`	*MSERVER_HOME*`/nodemanager`

Granting Privileges to the wlsifconfig.sh Script

Grant *sudo* privilege to the operating system user (for example, `oracle`) with no password restriction, and grant execute privilege on the `/sbin/ifconfig` and `/sbin/arping` binaries.

> **Note:**
>
> For security reasons, `sudo` should be restricted to the subset of commands required to run the `wlsifconfig.sh` script.
>
> Ask the system administrator for the *sudo* and system rights as appropriate to perform this required configuration task.

The following is an example of an entry inside */etc/sudoers* granting *sudo* execution privilege for `oracle` to run `ifconfig` and `arping`:

```
Defaults:oracle !requiretty
oracle ALL=NOPASSWD: /sbin/ifconfig,/sbin/arping
```

Configuring Server Migration Targets

To configure migration in a cluster:

1. Sign in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure window, expand **Environment** and select **Clusters**. The Summary of Clusters page is displayed.
3. Click the cluster for which you want to configure migration in the Name column of the table.
4. Click the **Migration** tab.
5. Click **Lock & Edit**.
6. Select **Database** as Migration Basis. From the drop-down list, select **Leasing** as Data Source For Automatic Migration.
7. Under **Candidate Machines For Migratable Server**, in the Available filed, select the Managed Servers in the cluster and click the right arrow to move them to **Chosen**.
8. Click **Save**.

9. Set the Candidate Machines for Server Migration. You must perform this task for all of the managed servers as follows:
   a. In Domain Structure window of the Oracle WebLogic Server Administration Console, expand **Environment** and select **Servers**.
   b. Select the server for which you want to configure migration.
   c. Click the **Migration** tab.
   d. Select **Automatic Server Migration Enabled** and click **Save**.

      This enables the Node Manager to start a failed server on the target node automatically.

      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 the host to which the Managed Server should failover if the current host is unavailable. For example, for the Managed Server on the HOST1, select HOST2; for the Managed Server on HOST2, select HOST1.

> **Tip:**
>
> Click **Customize this table** in the Summary of Servers page, move Current Machine from the Available Window to the Chosen window to view the machine on which the server is running. This is different from the configuration if the server is migrated automatically.

10. Click **Activate Changes**.
11. Restart the Administration Server and the servers for which server migration has been configured.

Testing Whole Server Migration

Perform the steps in this section to verify that automatic whole server migration is working properly.

To test from Node 1:

1. Stop the managed server process.

   ```
   kill -9 pid
   ```

   pid specifies the process ID of the managed server. You can identify the *pid* in the node by running this command:
2. Watch the Node Manager console (the terminal window where you performed the kill command): you should see a message indicating that the managed server's floating IP has been disabled.
3. Wait for the Node Manager to try a second restart of the Managed Server. Node Manager waits for a period of 30 seconds before trying this restart.

4. After node manager restarts the server and before it reaches *Running* state, kill the associated process again.

   Node Manager should log a message indicating that the server will not be restarted again locally.

> **Note:**
>
> The number of restarts required is determined by the `RestartMax` parameter in the following configuration file:
>
> The default value is `RestartMax=2`.

To test from Node 2:

1. Watch the local Node Manager console. After 30 seconds since the last try to restart the managed server on Node 1, Node Manager on Node 2 should prompt that the floating IP for the managed server is being brought up and that the server is being restarted in this node.
2. Access a product URL by using the same IP address. If the URL is successful, then the migration was successful.

Verification From the Administration Console

You can also verify migration using the Oracle WebLogic Server Administration Console:

1. Log in to the Administration Console.
2. Click **Domain** on the left console.
3. Click the **Monitoring** tab and then the **Migration** subtab.

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

> **Note:**
>
> After a server is migrated, to fail it back to its original machine, stop the managed server from the Oracle WebLogic Administration Console and then start it again. The appropriate Node Manager starts the managed server on the machine to which it was originally assigned.

15

Configuring Single Sign-On for an Enterprise Deployment

You need to configure the Oracle HTTP Server WebGate in order to enable single sign-on with Oracle Access Manager.

- About Oracle 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.
- General Prerequisites for Configuring Oracle HTTP Server WebGate
  Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.
- Enterprise Deployment Prerequisites for Configuring OHS 12c 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.
- Configuring Oracle HTTP Server 12c WebGate for an Enterprise Deployment
  You need to perform the following steps in order to configure Oracle HTTP Server 12*c* WebGate for Oracle Access Manager on both WEBHOST1 and WEBHOST2.
- Registering the Oracle HTTP Server WebGate with Oracle Access Manager
  You can register the WebGate agent with Oracle Access Manager using the Oracle Access Manager Administration console.
- Setting Up the WebLogic Server Authentication Providers
  To set up the WebLogic Server authentication providers, back up the configuration files, set up the Oracle Access Manager Identity Assertion Provider and set the order of providers.
- Configuring Oracle ADF and OPSS Security with Oracle Access Manager
  Some Oracle Fusion Middleware management consoles use Oracle Application Development Framework (Oracle ADF) security, which can integrate with Oracle Access Manager Single Sign On (SSO). These applications can take advantage of Oracle Platform Security Services (OPSS) SSO for user authentication, but you must first configure the domain-level `jps-config.xml` file to enable these capabilities.
- Configuring Single Sign-On for Applications
  This section describes how to enable single sign-on (SSO) for BI applications.

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 Oracle WebGate software is installed as part of the Oracle HTTP Server 12*c* software installation. See Registering and Managing OAM 11g Agents in *Adminstrator's Guide for Oracle Access Management*.

General Prerequisites for Configuring Oracle HTTP Server WebGate

Before you can configure Oracle HTTP Server WebGate, you must have installed and configured a certified version of Oracle Access Manager.

For the most up-to-date information, see the certification document for your release on the *Oracle Fusion Middleware Supported System Configurations* page.

For WebGate certification matrix, click and open http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/oam-webgates-2147084.html, then click the *Certification Matrix for 12c Access Management WebGates* link to download the certification matrix spreadsheet.

> **Note:**
>
> For production environments, it is highly recommended that you install Oracle Access Manager in its own environment and not on the machines that are hosting the enterprise deployment.

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.

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 of 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.

> **Note:**
>
> It is recommended that you use the WebGate version that is certified with your Oracle Access Manager deployment.

Configuring Oracle HTTP Server 12*c* WebGate for an Enterprise Deployment

You need to perform the following steps in order to configure Oracle HTTP Server 12*c* WebGate for Oracle Access Manager on both WEBHOST1 and WEBHOST2.

In the following procedure, replace the directory variables, such as *WEB_ORACLE_HOME* and *WEB_CONFIG_DIR*, with the values, as defined in File System and Directory Variables Used in This Guide.

1. Perform a complete backup of the web tier domain.
2. Change directory to the following location in the Oracle HTTP Server Oracle home:

   ```
   cd WEB_ORACLE_HOME/webgate/ohs/tools/deployWebGate/
   ```

3. Run the following command to create the WebGate Instance directory and enable WebGate logging on OHS Instance:

   ```
   ./deployWebGateInstance.sh -w WEB_CONFIG_DIR -oh WEB_ORACLE_HOME
   ```

4. Verify that a `webgate` directory and subdirectories was created by the `deployWebGateInstance` command:

   ```
   ls -lat WEB_CONFIG_DIR/webgate/
   total 16
   drwxr-x---+ 8 orcl oinstall 20 Oct  2 07:14 ..
   drwxr-xr-x+ 4 orcl oinstall  4 Oct  2 07:14 .
   drwxr-xr-x+ 3 orcl oinstall  3 Oct  2 07:14 tools
   drwxr-xr-x+ 3 orcl oinstall  4 Oct  2 07:14 config
   ```

5. Run the following command to ensure that the `LD_LIBRARY_PATH` environment variable contains *WEB_ORACLE_HOME*`/lib` directory path:

   ```
   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:WEB_ORACLE_HOME/lib
   ```

6. Change directory to the following directory

   ```
   WEB_ORACLE_HOME/webgate/ohs/tools/setup/InstallTools
   ```

7. Run the following command from the `InstallTools` directory.

   ```
   ./EditHttpConf -w WEB_CONFIG_DIR -oh WEB_ORACLE_HOME -o
   output_file_name
   ```

   > **Note:**
   >
   > The `-oh` *WEB_ORACLE_HOME* and `-o` *output_file_name* parameters are optional.

   This command:

- Copies the `apache_webgate.template` file from the Oracle HTTP Server Oracle home to a new `webgate.conf` file in the Oracle HTTP Server configuration directory.
- Updates the `httpd.conf` file to add one line, so it includes the `webgate.conf`.
- Generates a WebGate configuration file. The default name of the file is `webgate.conf`, but you can use a custom name by using the `-o` *`output_file_name`* argument to the command.

Registering the Oracle HTTP Server WebGate with Oracle Access Manager

You can register the WebGate agent with Oracle Access Manager using the Oracle Access Manager Administration console.

See Registering an OAM Agent Using the Console in *Administrator's Guide for Oracle Access Management*.

- 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
- Insert OHS SimpleCA Certificate into the Wallet Artifact
- Enable MD5 Certificate Signatures for the Oracle HTTP Server Instances
- Restarting the Oracle HTTP Server Instance

About RREG In-Band and Out-of-Band Mode

You can run the RREG Tool in one of the 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 to 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).
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.

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 take a default value.
> - In the primary server list, the default names are mentioned as OAM_SERVER1 and OAM_SERVER2 for OAM servers. Rename these names in the list if the server names are changed in your environment.

To perform this task:

1. If you are using in-band mode, then change directory to the following location on one of the OAM Servers:

   OAM_ORACLE_HOME/oam/server/rreg/input

   If you are using out-of-band mode, then change directory to the location where you unpacked the RREG archive on the WEBHOST1 server.
2. Make a copy of the `OAM11GRequest.xml` file template with an environment-specific name.

   ```
   cp OAM11GRequest.xml OAM11GRequest_edg.xml
   ```

3. Review the properties listed in the file, and then update your copy of the `OAM11GRequest.xml` file to make sure that the properties reference the host names and other values specific to your environment.

Table 15-1 Fields in the OAM11GRequest.xml file.

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 that 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`	Must be set to the security mode configured on the Oracle Access Management server. This is one of the three modes: open, simple, or certificate. **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. In most cases, 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 WebGates in *Administrator's Guide for Oracle Access Management*.
`cachePragmaHeader`	private
`cacheControlHeader`	private

Table 15-1 (Cont.) Fields in the OAM11GRequest.xml file.

OAM11gRequest.xml Property	Set to...
`ipValidation`	0 `<ipValidation>0</ipValidation>` If ipValidation is set to '1', the IP address stored in the cookie must match the client's IP address, otherwise, the SSO cookie is rejected and the user must reauthenticate. This can cause problems with certain Web applications. For example, Web applications managed by a proxy server typically change the user's IP address, substituting the IP address of the proxy. Setting to '0' Disables IP validation.

`ipValidationExceptions`	Can be empty when ipValidation is '0'.
	If IP Validation is true, the IP address is compared to the IP Validation Exceptions list. If the address is found on the exceptions list, it does not need to match the IP address stored in the cookie. You can add as many IP addresses as needed. For example, the IP address of the front end load balancer:
	
	`<ipValidationExceptions>`
	`<ipAddress>130.35.165.42</ipAddress>`
	`</ipValidationExceptions>`

`agentBaseUrl`	Fully-qualified URL with the host and the port of the front-end Load Balancer VIP in front of the WEBHOST*n* machines on which Oracle HTTP 12*c* WebGates are installed.
	For example:
	
	`<agentBaseUrl>`
	`https://bi.example.com:443`
	`</agentBaseUrl>`

`virtualHost`	Set to **true** when protecting more than the `agentBaseUrl`, such as SSO protection for the administrative VIP.

Table 15-1 (Cont.) Fields in the OAM11GRequest.xml file.

OAM11gRequest.xml Property	Set to...

`hostPortVariationsList`	Add `hostPortVariation` host and port elements for each of the load-balancer URLs that are protected by the WebGates.
	For example:
	
	`<hostPortVariationsList>`
	`<hostPortVariations>`
	`<host>biinternal.example.com</host>`
	`<port>80</port>`
	`</hostPortVariations>`
	`<hostPortVariations>`
	`<host>admin.example.com</host>`
	`<port>80</port>`
	`</hostPortVariations>`
	`<hostPortVariations>`
	`<host>osb.example.com</host>`
	`<port>443</port>`
	`</hostPortVariations>`
	`</hostPortVariationsList>`

`logOutUrls`	Leave it empty. The Logout URL triggers the logout handler, which removes the cookie and requires the user to re-authenticate the next time the user accesses a resource protected by Access Manager. If Logout URL is not configured, the request URL is checked for *logout*. and, if found (except *logout.gif* and *logout.jpg*), also triggers the logout handler. If a value is set to this property, all used logout URLs must be added.

`primaryServerList`	Verify that the host and the port of the OAM Managed Servers matches with this list. Example:
	
	`<primaryServerList>`
	`<Server>`
	`<host>wls_oam1</host>`
	`<port>14100</port>`
	`<numOfConnections>1</numOfConnections>`
	`</Server>`
	`<Server>`
	`<host>wls_oam2</host>`
	`<port>14100</port>`
	`<numOfConnections>2</numOfConnections>`
	`</Server>`
	`</primaryServerList>`

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 have not already opened the copied `OAM11GRequest_edg.xml` file for editing, locate, and open the file in a text editor.

   See 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 in this example for backward compatibility with previous versions of Oracle Access Manager.

```
<protectedResourcesList>
        <resource>/analytics</resource>
        <resource>/analytics/saw.dll</resource>
        <resource>/bicontent</resource>
        <resource>/xmlpserver</resource>
        <resource>/mapviewer</resource>
        <resource>/mapviewer/console</resource>
        <resource>/mapviewer/mapadmin</resource>
        <resource>/mapviewer/mcsadmin</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>
        <resource>/analytics/jbips</resource>
        <resource>/cds/</resource>
        <resource>/aps/SmartView/</resource>
</protectedResourcesList>
<publicResourcesList>
        <resource>/aps</resource>
        <resource>/aps/JAPI</resource>
        <resource>/mapviewer/dataserver</resource>
        <resource>/mapviewer/foi</resource>
```

```
        <resource>/mapviewer/mcserver</resource>
        <resource>/mapviewer/wms</resource>
        <resource>/mapviewer/wmts</resource>
        <resource>/aps/Essbase</resource>
        <resource>/essbase/agent</resource>
        <resource>/essbase-webservices</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_edg.xml` file.

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.

- Running the RREG Tool in In-Band Mode
- Running the RREG Tool in Out-Of-Band Mode

Running the RREG Tool in In-Band Mode

To run the RREG Tool in in-band mode:

1. Change to 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`

   If you are using out-of-band mode, then the RREG home directory is the location where you unpacked the RREG archive.

2. Change to the following directory:

   - (UNIX) *RREG_HOME*`/bin`
   - (Windows) *RREG_HOME*`\bin`

   `cd` *RREG_HOME*`/bin/`

3. Set the permissions of the `oamreg.sh` command so that you can execute the file:

   ```
   chmod +x oamreg.sh
   ```

4. Enter the following command:

```
./oamreg.sh inband RREG_HOME/input/OAM11GRequest_edg.xml
```

In this example:

- It is assumed that the edited `OAM11GRequest.xml` file is located in the *`RREG_HOME`*`/input` directory.
- The output from this command is 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/OAM11GRequest_edg.xml
Enter admin username:weblogic_idm
Username: weblogic_iam
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:SOA12213_EDG_AGENT
Base URL: https://soa.example.com:443
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.
```

Running the RREG Tool in Out-Of-Band Mode

To run the RREG Tool in out-of-band mode on the WEBHOST server, the 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/

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. See Securing Communication Between OAM Servers and WebGates in *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/ - For WebGate 11*g* (11.1.2.3). • *RREG_HOME*/output/Agent_ID/wallet - For WebGate 11*g* (11.1.2.2) and OHS 12*c*.
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/

File	Location
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.

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 Server configuration directory, based on the security mode setting for Oracle Access Manager. In some cases, you might have to create the directories if they do not exist already. For example, the wallet directory might not exist in the configuration directory.

> **Note:**
>
> For an enterprise deployment, Oracle recommends simple mode, unless additional requirements exist to implement custom security certificates for the encryption of authentication and authorization traffic. The information about using open or certification mode is provided here as a convenience.
>
> Avoid using open mode, because in open mode, traffic to and from the Oracle Access Manager server is not encrypted.
>
> For more information about using certificate mode or about Oracle Access Manager supported security modes in general, see Securing Communication Between OAM Servers and WebGates in *Administrator's Guide for Oracle Access Management*.

Table 15-2 Web Tier Host Location to Copy the Generated Artifacts

File	Location When Using OPEN Mode	Location When Using SIMPLE Mode	Location When Using CERT Mode
`wallet/cwallet.sso`[1]	*WEB_CONFIG_DIR*`/webgate/config/wallet`	*WEB_CONFIG_DIR*`/webgate/config/wallet/` By default the wallet folder is not available. Create the wallet folder under *WEB_CONFIG_DIR*`/webgate/config/`.	*WEB_CONFIG_DIR*`/webgate/config/wallet/`
`ObAccessClient.xml`	*WEB_CONFIG_DIR*`/webgate/config`	*WEB_CONFIG_DIR*`/webgate/config/`	*WEB_CONFIG_DIR*`/webgate/config/`
`password.xml`	N/A	*WEB_CONFIG_DIR*`/webgate/config/`	*WEB_CONFIG_DIR*`/webgate/config/`
`aaa_key.pem`	N/A	*WEB_CONFIG_DIR*`/webgate/config/simple/`	*WEB_CONFIG_DIR*`/webgate/config/`
`aaa_cert.pem`	N/A	*WEB_CONFIG_DIR*`/webgate/config/simple/`	*WEB_CONFIG_DIR*`/webgate/config/`

[1] Copy `cwallet.sso` from the wallet folder and not from the output folder. Even though there are 2 files with the same name they are different. The one in the wallet sub directory is the correct one.

> **Note:**
>
> If you need to redeploy the `ObAccessClient.xml` to `WEBHOST1` and `WEBHOST2`, delete the cached copy of `ObAccessClient.xml` and its lock file, `ObAccessClient.xml.lck` from the servers. The cache location on `WEBHOST1` is:
>
> *WEB_DOMAIN_HOME*`/servers/ohs1/cache/`
>
> And you must perform the similar step for the second Oracle HTTP Server instance on `WEBHOST2`:
>
> *WEB_DOMAIN_HOME*`/servers/ohs2/cache/`

Insert OHS SimpleCA Certificate into the Wallet Artifact

If the OHS servers have been configured with an 11*g* or earlier version of the OAM server, there is a need to insert the OHS SimpleCA certificate into the wallet file artifact that was deployed in Copying Generated Artifacts to the Oracle HTTP Server WebGate Instance Location.

Complete the following steps:

1. On WEBHOST1, go to the following directory:

   ```
   WEB_CONFIG_DIR/webgate/config/wallet
   ```

2. Run the following command to insert the SimpleCA certificate into the wallet file:

```
WEB_ORACLE_HOME/oracle_common/bin/orapki wallet add -wallet ./ -
trusted_cert -cert WEB_ORACLE_HOME/webgate/ohs/tools/openssl/simpleCA/
cacert.pem -auto_login_only
```

The following output is displayed:

```
  simpleCA/cacert.pem -auto_login_only
  Oracle PKI Tool : Version 12.2.1.3.0
 Copyright (c) 2004, 2017, Oracle and/or its affiliates. All rights
reserved.

 Operation is successfully completed.
```

3. Validate the certificate insertion with the following command:

```
WEB_ORACLE_HOME/oracle_common/bin/orapki wallet display -wallet ./
```

The following output is displayed:

```
  Oracle PKI Tool : Version 12.2.1.3.0
  Copyright (c) 2004, 2017, Oracle and/or its affiliates. All rights
reserved.

  Requested Certificates:
  User Certificates:
  Oracle Secret Store entries: OAMAgent@#3#@wcedgRwse01Env1Ps3_Key
  Trusted Certificates:
  Subject: CN=NetPoint Simple Security CA - Not for General
Use,OU=NetPoint,O=Oblix\, Inc.,L=Cupertino,ST=California,C=US
```

4. Repeat steps 1 through 3 on WEBHOST2.

Enable MD5 Certificate Signatures for the Oracle HTTP Server Instances

Some releases of Oracle Access Management Server implement simple mode security certificates by using MD5 signatures unless upgraded or patched appropriately. Oracle Recommends that, if possible, the OAM certificates are upgraded to SHA-2 certificates. This might not be possible for customers who have several versions of Oracle HTTP server to contend with.

If upgrading the certificates is not possible, support for MD5 signatures must be enabled manually to make Oracle HTTP server 12.2.1.*x* work with Oracle Access Manager 11*g*'s MD5 certificates when you use a webgate in simple security mode.

To enable MD5 certificate signatures on each OHS instance, complete the following steps:

1. On WEBHOST1, change to the following directory:

   WEB_DOMAIN_HOME`/config/fmwconfig/components/OHS/ohs1`

2. Open the `ohs.plugins.nodemanager.properties` file, add the following line, and save the file.

   ```
   environment.ORACLE_SSL_ALLOW_MD5_CERT_SIGNATURES = 1
   ```

3. Repeat steps 1 and 2 for all other instances on the WEBHOST*n* servers.

   For example, the `ohs2` instance on WEBHOST2

> **Note:**
>
> The change takes effect when the instances are restarted in the next topic.

Restarting the Oracle HTTP Server Instance

For information about restarting the Oracle HTTP Server instance, see Restarting Oracle HTTP Server Instances by Using WLST in *Administering Oracle HTTP Server*.

If you have configured Oracle HTTP Server in a WebLogic Server domain, you can also use Oracle Fusion Middleware Control to restart the Oracle HTTP Server instances. See Restarting Oracle HTTP Server Instances by Using Fusion Middleware Control in *Administering Oracle HTTP Server*.

Setting Up the WebLogic Server Authentication Providers

To set up the WebLogic Server authentication providers, back up the configuration files, set up the Oracle Access Manager Identity Assertion Provider and set the order of providers.

The following topics assumes that you have already configured the LDAP authenticator by following the steps in Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group. If you have not already created the LDAP authenticator, then do so before you continue with this section.

- Backing Up Configuration Files
- Setting Up the Oracle Access Manager Identity Assertion Provider
- Updating the Default Authenticator and Setting the Order of Providers

Backing Up Configuration Files

To be safe, you should first back up the relevant configuration files:

```
ASERVER_HOME/config/config.xml
ASERVER_HOME/config/fmwconfig/jps-config.xml
ASERVER_HOME/config/fmwconfig/system-jazn-data.xml
```

Also back up the `boot.properties` file for the Administration Server:

```
ASERVER_HOME/servers/AdminServer/security/boot.properties
```

Setting Up the Oracle Access Manager Identity Assertion Provider

Set up an Oracle Access Manager identity assertion provider in the Oracle WebLogic Server Administration Console.

To set up the Oracle Access Manager identity assertion provider:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. Click **Security Realms** in the left navigation bar.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. Click **New**, and select the asserter type **OAMIdentityAsserter** from the drop-down menu.
7. Name the asserter (for example, *OAM ID Asserter*), and click **OK**.
8. Click the newly added asserter to see the configuration screen for the Oracle Access Manager identity assertion provider.
9. Set the control flag to *REQUIRED*.
10. Under Chosen types, select both the **ObSSOCookie** and **OAM_REMOTE_USER** options, if they are not selected by default.
11. Click **Save** to save the settings.
12. Click **Activate Changes** to propagate the changes.

Updating the Default Authenticator and Setting the Order of Providers

Set the order of identity assertion and authentication providers in the WebLogic Server Administration console.

To update the default authenticator and set the order of the providers:

1. Log in to the WebLogic Server Administration Console, if not already logged in.
2. Click **Lock & Edit**.
3. From the left navigation, select **Security Realms**.
4. Click the **myrealm** default realm entry.
5. Click the **Providers** tab.
6. From the table of providers, click the **DefaultAuthenticator**.
7. Set the Control Flag to `SUFFICIENT`.
8. Click **Save** to save the settings.
9. From the navigation breadcrumbs, click **Providers** to return to the list of providers.
10. Click **Reorder**.
11. Sort the providers to ensure that the OAM Identity Assertion provider is first and the DefaultAuthenticator provider is last.

Table 15-3 Sort order

Sort Order	Provider	Control Flag
1	OAMIdentityAsserter	REQUIRED
2	LDAP Authentication Provider	SUFFICIENT
3	DefaultAuthenticator	SUFFICIENT
4	Trust Service Identity Asserter	N/A
5	DefaultIdentityAsserter	N/A

12. Click **OK**.
13. Click **Activate Changes** to propagate the changes.
14. Shut down the Administration Server, Managed Servers, and any system components, as applicable.
15. Restart the Administration Server.
16. If you are going to configure ADF consoles with SSO, you can keep the managed servers down and restart them later. If not, you need to restart managed servers now.

Configuring Oracle ADF and OPSS Security with Oracle Access Manager

Some Oracle Fusion Middleware management consoles use Oracle Application Development Framework (Oracle ADF) security, which can integrate with Oracle Access Manager Single Sign On (SSO). These applications can take advantage of Oracle Platform Security Services (OPSS) SSO for user authentication, but you must first configure the domain-level `jps-config.xml` file to enable these capabilities.

The domain-level `jps-config.xml` file is located in the following location after you create an Oracle Fusion Middleware domain:

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

> **Note:**
>
> The domain-level `jps-config.xml` should not be confused with the `jps-config.xml` that is deployed with custom applications.

To update the OPSS configuration to delegate SSO actions in Oracle Access Manager, complete the following steps:

1. Change to the following directory:

   ORACLE_COMMON_HOME`/common/bin`

2. Start the WebLogic Server Scripting Tool (WLST):

   ```
   ./wlst.sh
   ```

3. Connect to the Administration Server, by using the following WLST command:

   ```
   connect('admin_user','admin_password','admin_url')
   ```

   For example:

   ```
   connect('weblogic_bi','mypassword','t3://ADMINVHN:7001')
   ```

4. Run the `addOAMSSOProvider` command, as shown:

   ```
   addOAMSSOProvider(loginuri="/${app.context}/
   adfAuthentication", logouturi="/oamsso/logout.html")
   ```

   The following table defines the expected value for each argument in the `addOAMSSOProvider` command.

Table 15-4 Expected Values for the Argument in the `addOAMSSOProvider` command

Argument	Definition
loginuri	Specifies the URI of the login page **Note:** For ADF security enabled applications, "/*context-root*/adfAuthentication" should be provided for the 'loginuri' parameter. For example: `/${app.context}/adfAuthentication` **Note:** `${app.context}` must be entered as shown. At runtime, the application replaces the variable appropriately. Here is the flow: a. User accesses a resource that has been protected by authorization policies in OPSS, fox example. b. If the user is not yet authenticated, ADF redirects the user to the URI configured in *loginuri*. c. Access Manager, should have a policy to protect the value in *loginuri*: for example, "/*context-root*/adfAuthentication". d. When ADF redirects to this URI, Access Manager displays a Login Page (depending on the authentication scheme configured in Access Manager for this URI).
logouturi	Specifies the URI of the logout page. The value of the *loginurl* is usually `/oam/logout.html`.

Table 15-4 (Cont.) Expected Values for the Argument in the `addOAMSSOProvider` command

Argument	Definition
autologinuri	Specifies the URI of the autologin page. This is an optional parameter.

5. Disconnect from the Administration Server by entering the following command:

   ```
   disconnect()
   ```

6. Restart the Administration Server and the managed servers.

Configuring Single Sign-On for Applications

This section describes how to enable single sign-on (SSO) for BI applications.

It includes the following topics.

- Enabling Single Sign-On and Oracle Access Manager for Oracle BI EE
- Enabling Single Sign-On and Oracle Access Manager for Oracle BI Publisher

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.

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://BIHOST1:7003/xmlpserver
   - http://BIHOST2:7003/xmlpserver

   You will be redirected to:

   `http://bi.example.com/xmlpserver`
2. In BI Publisher, click **Administration**, then **Security Configuration**.
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.

   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 that you use GridLink data sources when you develop 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.

- About Multi Data Sources and Oracle RAC
  A multi data source provides an ordered list of data sources to use to satisfy connection requests.
- Typical Procedure for Configuring Multi Data Sources for an Enterprise Deployment
  You need to 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.

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 *Administering JDBC Data Sources for Oracle WebLogic Server*.

Typical Procedure for Configuring Multi Data Sources for an Enterprise Deployment

You need to 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.

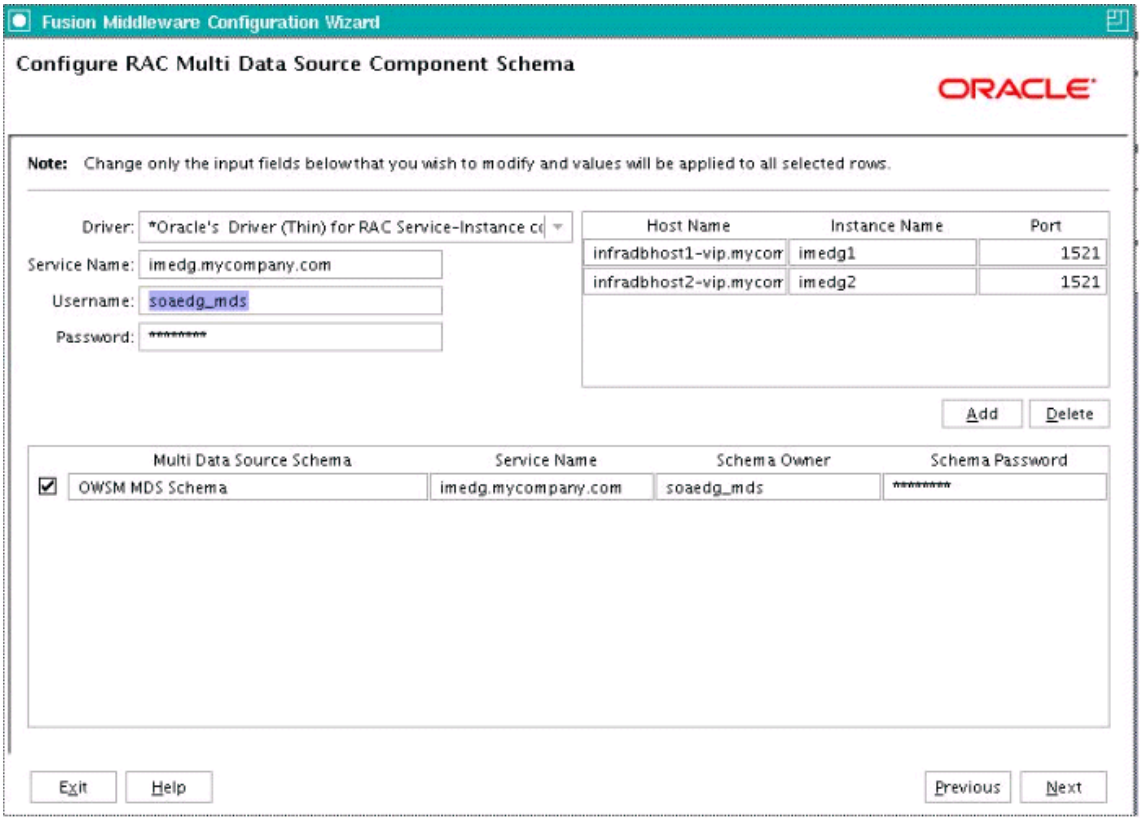
b. For the RAC configuration for component schemas, **Convert to RAC multi data source**.

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

D

data sources, *A-2*

R

RAC database, *A-2*

V

virtual server, *5-2*