

Oracle® Fusion Middleware

Enterprise Deployment Guide for Oracle SOA Suite



14c (14.1.2.0.0)

F85479-01

February 2025

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite, 14c (14.1.2.0.0)

F85479-01

Copyright © 2014, 2025, Oracle and/or its affiliates.

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, software documentation, data (as defined in the Federal Acquisition Regulation), 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 embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside 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, Epyc, and the AMD 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	xviii
Documentation Accessibility	xviii
Diversity and Inclusion	xviii
Conventions	xviii

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	2-2
Understanding the Firewalls and Zones of a Typical Enterprise Deployment	2-3
Understanding the Elements of a Typical Enterprise Deployment Topology	2-3
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	2-8
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	2-9
Using Oracle Web Services Manager in the Application Tier	2-10
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	2-11
About Using Unicast for Communications within the Application Tier	2-12
Understanding OPSS and Requests to the Authentication and Authorization Stores	2-13
About Coherence Clusters In a Typical Enterprise Deployment	2-14
About the Data Tier	2-15

3 About the Oracle SOA Suite Enterprise Deployment Topology

About the Primary and Build-Your-Own Enterprise Deployment Topologies	3-2
Diagrams of the Primary Oracle SOA Suite Enterprise Topologies	3-2
Diagram of the Oracle SOA Suite and Oracle Service Bus Topology	3-2
Diagram of the Oracle SOA Suite and Oracle Business Activity Monitoring Topology	3-4
About the Primary Oracle SOA Suite Topology Diagrams	3-6
About the Topology Options for Oracle Service Bus	3-6
Summary of Oracle SOA Suite Load Balancer Virtual Server Names	3-6
About the Routing of SOA Composite Requests	3-8
More About the soainternal Virtual Server Name	3-8
About Web Services Optimizations for SOA Composite Applications	3-8
About Accessing SOA Composite Applications through Oracle HTTP Server	3-9
About Accessing Oracle SOA Suite Composite Applications Through the Load Balancer	3-10
Summary of the Managed Servers and Clusters on SOA Application Tier	3-10
Flow Charts and Road Maps for Implementing the Primary Oracle SOA Suite Enterprise Topologies	3-11
Flow Chart of the Steps to Install and Configure the Primary Oracle SOA Suite Enterprise Topologies	3-11
Roadmap Table for Planning and Preparing for an Enterprise Deployment	3-12
Roadmap Table for Configuring the Oracle SOA Suite and Oracle Service Bus Enterprise Topology	3-13
Roadmap Table for Configuring the Oracle SOA Suite and Oracle Business Activity Monitoring Enterprise Topology	3-14
Building Your Own Oracle SOA Suite Enterprise Topology	3-15
Flow Chart of the Build Your Own Enterprise Topologies	3-15
Description of the Supported Build Your Own Topologies	3-16
About Installing and Configuring a Custom Enterprise Topology	3-18
About Using SSL Certificates in the Oracle SOA Suite Enterprise Topology	3-18
About Using JDBC Persistent Stores	3-19
About Using Automatic Service Migration for the Oracle SOA Suite Enterprise Topology	3-19
About Reference Configuration for SOA and OSB	3-19

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	4-1
Using the Oracle SOA Suite Enterprise Deployment Workbook	4-2
Locating the Oracle SOA Suite Enterprise Deployment Workbook	4-2
Understanding the Contents of the Oracle SOA Suite Enterprise Deployment Workbook	4-2
Using the Start Tab	4-2
Using the Hardware - Host Computers Tab	4-3
Using the Network - Virtual Hosts & Ports Tab	4-3
Using the Storage - Directory Variables Tab	4-4
Using the Database - Connection Details Tab	4-4
Who Should Use the Enterprise Deployment Workbook?	4-4

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	5-3
Typical Disk Space Requirements for an Enterprise Deployment	5-5
Operating System Requirements for an Enterprise Deployment Topology	5-5
Reserving the Required IP Addresses for an Enterprise Deployment	5-6
What is a Virtual IP (VIP) Address?	5-6
Why Use Virtual Host Names and Virtual IP Addresses?	5-7
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	6-3
Additional Instructions for admin.example.com	6-4
Additional Instructions for soa.example.com	6-4
Additional Instructions for soainternal.example.com	6-4
Additional Instructions for osb.example.com	6-4
Additional Instructions for mft.example.com	6-5

Configuring the Firewalls and Ports for an Enterprise Deployment	6-5
--	-----

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	8-1
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	8-3
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	8-6
Configuring a Host to Use an NIS/YP Host	8-7
Mounting the Required Shared File Systems on Each Host	8-8
Enabling the Required Virtual IP Addresses on Each Host	8-10

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
Setting the PROCESSES Database Initialization Parameter for an Enterprise Deployment	9-3
Creating Database Services	9-4
Using SecureFiles for Large Objects (LOBs) in an Oracle Database	9-6
About Database Backup Strategies	9-7
Implementing a Database Growth Management Strategy for Oracle SOA Suite	9-7

10 Creating the Initial Infrastructure Domain for an Enterprise Deployment

About the Initial Infrastructure Domain	10-2
About the Infrastructure Distribution	10-2
Characteristics of the Domain	10-2
Variables Used When Creating the Infrastructure Domain	10-3
Installing the Oracle Fusion Middleware Infrastructure on SOAHOST1	10-3
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 SOAHOST1	10-5
Navigating the Infrastructure Installation Screens	10-6
Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers	10-7
Checking the Directory Structure	10-8
Disabling the Derby Database	10-8
Creating the Database Schemas	10-9
Installing and Configuring a Certified Database	10-9
Starting the Repository Creation Utility (RCU)	10-9
Navigating the RCU Screens to Create the Schemas	10-10
Verifying Schema Access	10-13
Configuring the Infrastructure Domain	10-13
Starting the Configuration Wizard	10-14
Navigating the Configuration Wizard Screens to Configure the Infrastructure Domain	10-14
Download and Configure WebLogic Remote Console	10-24
Configuring SSL Certificates for the Domain	10-25
Creating Certificates and Certificate Stores for the WebLogic Domain	10-25
Adding Certificate Stores Location to the WebLogic Servers Start Scripts	10-26
Update Server's Security Settings Using the Remote Console	10-27
Connecting to the Remote Console Using the Administration Server's Virtual Hostname as Provider	10-27
Updating the WebLogic Servers Security Settings	10-28
Configuring KSS with Per-domain CA	10-30
Configuring a Per Host Node Manager for an Enterprise Deployment	10-31
Creating a Per Host Node Manager Configuration	10-31
Starting the Node Manager on SOAHOST1	10-33
Configuring the Node Manager Credentials	10-35
Enrolling the Domain with NM	10-35
Adding Truststore Configuration to Node Manager	10-36
Configuring the Domain Directories and Starting the Servers on SOAHOST1	10-37
Starting the Administration Server Using the Node Manager	10-37

Validating the Administration Server	10-39
Creating a Separate Domain Directory for Managed Servers on SOAHOST1	10-39
Starting and Validating the WLS_WSM1 Managed Server on SOAHOST1	10-41
Configuring Web Services Manager	10-42
Updating WebServices Domain Configuration	10-42
Bootstrapping WSM	10-42
Propagating the Domain and Starting the Servers on SOAHOST2	10-44
Unpacking the Domain on SOAHOST2	10-44
Starting the Node Manager on SOAHOST2	10-45
Starting and Validating the WLS_WSM2 Managed Server on SOAHOST2	10-46
Modifying the Upload and Stage Directories to an Absolute Path	10-46
Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group	10-46
About the Supported Authentication Providers	10-46
About the Enterprise Deployment Users and Groups	10-47
About Using Unique Administration Users for Each Domain	10-47
About the Domain Connector User	10-47
About Adding Users to the Central LDAP Directory	10-48
About Product-Specific Roles and Groups for Oracle SOA Suite	10-49
Example Users and Groups Used in This Guide	10-49
Prerequisites for Creating a New Authentication Provider and Provisioning Users and Groups	10-50
Provisioning a Domain Connector User in the LDAP Directory	10-50
Creating the New Authentication Provider	10-51
Provisioning an Enterprise Deployment Administration User and Group	10-55
Adding the Administration Role to the New Administration Group	10-56
Adding the wsm-pm Role to the Administrators Group	10-57
Backing Up the Configuration	10-57
Verification of Manual Failover of the Administration Server	10-58

11 Configuring Oracle HTTP Server for an Enterprise Deployment

About the Oracle HTTP Server Domains	11-2
Variables Used When Configuring the Oracle HTTP Server	11-2
Installing Oracle HTTP Server on WEBHOST1	11-2
Installing a Supported JDK	11-2
Locating and Downloading the JDK Software	11-3
Installing the JDK Software	11-3
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	11-8
Installing and Configuring an Oracle HTTP Server Domain on WEBHOST2	11-10
Starting the Node Manager and Oracle HTTP Server Instances on WEBHOST1 and WEBHOST2	11-10
Starting the Node Manager on WEBHOST1 and WEBHOST2	11-11
Starting the Oracle HTTP Server Instances	11-12
Setting Frontend Addresses and WebLogic Plugin for the WSM_PM Cluster and the Administration Server	11-12
Generate Required Certificates for OHS SSL Listeners	11-13
Configuring Oracle HTTP Server to Route Requests to the Application Tier	11-15
About the Oracle HTTP Server Configuration for an Enterprise Deployment	11-15
Purpose of the Oracle HTTP Server Virtual Hosts	11-15
About the WebLogicCluster Parameter of the <VirtualHost> Directive	11-15
Recommended Structure of the Oracle HTTP Server Configuration Files	11-16
Modifying the httpd.conf File to Include Virtual Host Configuration Files	11-16
Creating the Virtual Host Configuration Files	11-18
Validating the Virtual Server Configuration on the Load Balancer	11-22
Validating Access to the Management Consoles and Administration Server	11-22
Configure a New Provider in the WebLogic Remote Console to Access the Domain Configuration Through the Frontend LBR	11-23

12 Extending the Domain with Oracle SOA Suite

Variables Used When Configuring Oracle SOA Suite	12-2
Synchronizing the System Clocks	12-2
Installing the Software for an Enterprise Deployment	12-3
Starting the Oracle SOA Suite Installer on SOAHOST1	12-3
Navigating the Installation Screens	12-3
Installing Oracle SOA Suite on the Other Host Computers	12-4
Verifying the Installation	12-4
Reviewing the Installation Log Files	12-4
Checking the Directory Structure	12-4
Viewing the Contents of Your Oracle Home	12-5
Creating the Oracle SOA Suite Database Schemas	12-5
Starting the Repository Creation Utility (RCU)	12-5
Navigating the RCU Screens to Create the Schemas	12-6
Verifying Schema Access	12-8
Configuring SOA Schemas for Transactional Recovery	12-9
Extending the Enterprise Deployment Domain with Oracle SOA Suite	12-10
Starting the Configuration Wizard	12-10
Navigating the Configuration Wizard Screens to Extend the Domain with Oracle SOA Suite	12-11
Targeting Adapters Manually	12-17

Update Certificates for New Frontend Addresses	12-18
Update the WebLogic Servers Security Settings	12-18
Propagating the Extended Domain to the Domain Directories and Machines	12-18
Packing Up the Extended Domain on SOAHOST1	12-19
Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1	12-20
Unpacking the Domain on SOAHOST2	12-21
Starting and Validating the WLS_SOA1 Managed Server	12-22
Starting the WLS_SOA1 Managed Server	12-23
Adding the SOAAdmin Role to the Administrators Group	12-23
Validating the Managed Server by Logging in to the SOA Infrastructure	12-23
Starting and Validating the WLS_SOA2 Managed Server	12-24
Modifying the Upload and Stage Directories to an Absolute Path	12-24
Configuring the Web Tier for the Extended Domain	12-24
Generate the Required Certificates for OHS SSL Listeners	12-24
Configuring Oracle HTTP Server for the WLS_SOA Managed Servers	12-24
Validating the Oracle SOA Suite URLs Through the Load Balancer	12-27
Post-Configuration Steps for Oracle SOA Suite	12-28
Configuring Oracle Adapters for Oracle SOA Suite	12-28
Enabling High Availability for Oracle File and FTP Adapters	12-28
Enabling High Availability for Oracle JMS Adapters	12-31
Enabling High Availability for the Oracle Database Adapter	12-32
Considerations for Sync-Async Interactions in a SOA Cluster	12-33
Updating FusionAppsFrontendHostUrl	12-33
Replacing Connect Strings with the Appropriate TNS Alias	12-33
Backing Up the Configuration	12-33

13 Extending the Domain with Oracle Service Bus

About Configuring Oracle Service Bus in Its Own Domain	13-2
Variables Used When Configuring Oracle Service Bus	13-2
Overview of Adding OSB to the Topology	13-3
Prerequisites for Extending the Domain to Include Oracle Service Bus	13-4
Installing Oracle Service Bus Software	13-5
Starting the Oracle Service Bus Installer	13-5
Navigating the OSB Installation Screens	13-5
Installing the Software on Other Host Computers	13-6
Validating the OSB Installation	13-6
Reviewing the Installation Log Files	13-6
Checking the Directory Structure	13-7
Viewing the Contents of Your Oracle Home	13-7
Extending the SOA or Infrastructure Domain to Include Oracle Service Bus	13-7
Starting the Configuration Wizard	13-8

Navigating the Configuration Wizard Screens to Extend the Domain with Oracle Service Bus	13-8
Update Certificates for New Frontend Addresses	13-15
Update the WebLogic Servers Security Settings	13-15
Propagating the Extended Domain to the Domain Directories and Machines	13-15
Summary of the Tasks Required to Propagate the Changes to the Other Domain Directories and Machines	13-16
Starting and Validating the WLS_OSB1 Managed Server	13-16
Starting the WLS_OSB1 Managed Server	13-16
Adding the MiddlewareAdministrators Role to the Enterprise Deployment Administration Group	13-17
Validating the Managed Server	13-17
Starting and Validating the WLS_OSB2 Managed Server	13-17
Verifying the Appropriate Targeting and Configuration for OSB Singleton Services	13-18
Modifying the Upload and Stage Directories to an Absolute Path	13-19
Configuring the Web Tier for the Extended Domain	13-19
Generate the Required Certificates for OHS SSL Listeners	13-20
Configuring Oracle HTTP Server for the Oracle Service Bus	13-20
Validating the Oracle Service Bus URLs Through the Load Balancer	13-23
Post-Configuration Tasks for Oracle Service Bus	13-24
Enabling High Availability for Oracle DB, File and FTP Adapters	13-24
Considerations for Poller Transports	13-24
Configuring Specific Oracle Service Bus Services for an Enterprise Deployment	13-25
Replacing Connect Strings with the Appropriate TNS Alias	13-26
Backing Up the Configuration	13-26

14 Extending the Domain with Business Process Management

Variables Used When Configuring Business Process Management	14-2
Support for Reference Configuration in Business Process Management	14-2
Prerequisites for Extending the SOA Domain to Include Oracle BPM	14-3
Installing Oracle Business Process Management for an Enterprise Deployment	14-3
Starting the Installation Program	14-3
Navigating the Oracle BPM Installation Screens	14-4
Installing the Software on Other Host Computers	14-5
Verifying the Installation	14-6
Reviewing the Installation Log Files	14-6
Checking the Directory Structure	14-6
Viewing the Contents of Your Oracle Home	14-6
Running the Configuration Wizard on SOAHOST1 to Extend a SOA Domain to Include BPM	14-6
Starting the Configuration Wizard	14-7
Navigating the Configuration Wizard Screens to Extend the Domain with BPM	14-7
Propagating the Extended Domain to the Domain Directories and Machines	14-10

Starting the WLS_SOA Managed Servers with Business Process Management	14-10
Adding the Enterprise Deployment Administration User to the Oracle BPM Administrators Group	14-11
Configuring the Web Tier for the Extended Domain	14-11
Configuring Oracle HTTP Server for Oracle Business Process Management	14-11
Validating Access to Business Process Management Through the Hardware Load Balancer	14-12
Configuring BPMJMSModule for the Oracle BPM Cluster	14-14
Replacing Connect Strings with the Appropriate TNS Alias	14-15
Backing Up the Configuration	14-16

15 Extending the Domain with Oracle Enterprise Scheduler

About Adding Oracle Enterprise Scheduler	15-2
Variables Used When Configuring Oracle Enterprise Scheduler	15-3
Support for Reference Configuration in Oracle Enterprise Scheduler	15-4
Creating the Database Schemas for ESS	15-4
Starting the Repository Creation Utility (RCU)	15-4
Navigating the RCU Screens to Create the Enterprise Scheduler Schemas	15-5
Verifying Schema Access	15-7
Extending the SOA Domain to Include Oracle Enterprise Scheduler	15-8
Starting the Configuration Wizard	15-8
Navigating the Configuration Wizard Screens to Extend the Domain with Oracle Enterprise Scheduler	15-8
Update Certificates for New Frontend Addresses	15-14
Update the WebLogic Servers Security Settings	15-14
Propagating the Extended Domain to the Domain Directories and Machines	15-14
Adding the ESSAdmin Role to the SOA Administrators Group	15-15
Starting and Validating the WLS_ESS1 Managed Server	15-15
Starting and Validating the WLS_ESS2 Managed Server	15-16
Modifying the Upload and Stage Directories to an Absolute Path	15-17
Configuring the Web Tier for the Extended Domain	15-17
Configuring Oracle HTTP Server for the WLS_ESS Managed Servers	15-17
Validating Access to Oracle Enterprise Scheduler Through the Hardware Load Balancer	15-18
Replacing Connect Strings with the Appropriate TNS Alias	15-18
Backing Up the Configuration	15-19

16 Extending the Domain with Business Activity Monitoring

Variables Used When Configuring Business Activity Monitor	16-2
Support for Reference Configuration in BAM	16-3
About Configuring BAM in Its Own Domain	16-3
Prerequisites When Adding Oracle BAM to the Domain	16-3
Understanding the Installation Requirements for Adding Oracle BAM to the Domain	16-4

Understanding the Database Schema Requirements for Oracle BAM	16-4
Backing Up the Existing Installation	16-4
Special Instructions When Configuring Oracle BAM on Separate Hosts	16-4
Procuring Additional Host Computers for Oracle BAM	16-5
Installation Requirements When Configuring Oracle BAM on Separate Hosts	16-5
Installation Requirements When Using a Separate Volume or Partition	16-5
Installation Requirements When Using a Shared Oracle Home	16-6
Configuration Wizard Instructions When Configuring Oracle BAM on Separate Hosts	16-6
Propagating the Domain Configuration When Configuring Oracle BAM on Separate Hosts	16-6
Roadmap for Adding Oracle BAM to the Domain	16-7
Extending the SOA Domain to Include Oracle Business Activity Monitoring	16-8
Starting the Configuration Wizard	16-8
Navigating the Configuration Wizard Screens for Oracle BAM	16-8
Update Certificates for New Frontend Addresses	16-13
Update the WebLogic Servers Security Settings	16-14
Propagating the Extended Domain to the Domain Directories and Machines	16-14
Adding the Enterprise Deployment Administration User to the Oracle BAM Administration Group	16-15
Starting and Validating the WLS_BAM1 Managed Server	16-15
Starting and Validating the WLS_BAM2 Managed Server	16-16
Modifying the Upload and Stage Directories to an Absolute Path	16-17
Configuring the Web Tier for the Extended Domain	16-17
Configuring Oracle HTTP Server for the WLS_BAM Managed Servers	16-17
Validating Access to Oracle BAM Through the Hardware Load Balancer	16-18
Replacing Connect Strings with the Appropriate TNS Alias	16-18
Backing Up the Configuration	16-18

17 Extending the Domain with Oracle B2B

Variables Used When Configuring Oracle B2B	17-2
Support for Reference Configuration in Oracle B2B	17-2
Prerequisites for Extending the SOA Domain to Include Oracle B2B	17-3
Installing Oracle B2B for an Enterprise Deployment	17-3
Starting the Oracle B2B and Healthcare Installer on SOAHOST1	17-3
Navigating the Oracle B2B Installation Screens	17-4
Installing the Software on Other Host Computers	17-5
Verifying the B2B or Healthcare Installation	17-5
Reviewing the Installation Log Files	17-6
Viewing the Contents of Your Oracle Home	17-6
Running the Configuration Wizard to Extend for Oracle B2B	17-6
Starting the Configuration Wizard	17-6
Navigating the Configuration Wizard Screens for Oracle B2B	17-7

Propagating the Extended Domain to the Domain Directories and Machines	17-9
Starting the B2B Suite Components	17-10
Updating the B2B Instance Identifier for Transports	17-10
Configuring the Web Tier for the Extended Domain	17-11
Configuring Oracle HTTP Server for Oracle B2B	17-11
Adding the B2BAdmin Role to the SOA Administrators Group	17-12
Validating Access to Oracle B2B Through the Load Balancer	17-12
Replacing Connect Strings with the Appropriate TNS Alias	17-13
Backing Up the Configuration	17-13

18 Configuring Oracle Managed File Transfer in an Enterprise Deployment

About Oracle Managed File Transfer	18-2
About Managed File Transfer in an Enterprise Deployment	18-2
Characteristics of the Managed File Transfer Domain	18-4
Variables Used When Configuring Managed File Transfer	18-4
Synchronizing the System Clocks	18-5
Prerequisites for Creating the Managed File Transfer Domain	18-5
Installing the Software for an Enterprise Deployment	18-6
Starting the Managed File Transfer Installer on MFTHOST1	18-6
Navigating the Installation Screens When Installing Managed File Transfer	18-6
Installing the Software on Other Host Computers	18-7
Verifying the Installation	18-7
Reviewing the Installation Log Files	18-7
Checking the Directory Structure for Managed File Transfer	18-7
Creating the Managed File Transfer Database Schemas	18-8
Starting the Repository Creation Utility (RCU)	18-8
Navigating the RCU Screens to Create the Managed File Transfer Schemas	18-9
Verifying Schema Access	18-11
Creating the Managed File Transfer Domain for an Enterprise Deployment	18-12
Starting the Configuration Wizard	18-12
Navigating the Configuration Wizard Screens to Configure the MFT Domain	18-12
Download and Configure Weblogic Remote Console	18-22
Configuring SSL Certificates for the Domain	18-23
Creating Certificates and Certificate Stores for the WebLogic Domain	18-23
Adding Certificate Stores Location to the WebLogic Servers Start Scripts	18-24
Update Server's Security Settings Using the Remote Console	18-25
Connecting to the Remote Console Using the Administration Server's Virtual Hostname as Provider	18-25
Updating the WebLogic Servers Security Settings	18-26
Configuring KSS with Per-domain CA	18-27
Configuring a Per Host Node Manager for an Enterprise Deployment	18-28

Creating a Per Host Node Manager Configuration	18-29
Starting the Node Manager on MFTHOST1	18-31
Configuring the Node Manager Credentials	18-32
Enrolling the Domain with NM	18-33
Adding Truststore Configuration to Node Manager	18-34
Configuring the Domain Directories and Starting the Servers on MFTHOST1	18-34
Disabling the Derby Database	18-35
Starting the Administration Server Using the Node Manager	18-35
Validating the Administration Server	18-37
Creating a Separate Domain Directory for Managed Servers on MFTHOST1	18-37
Starting and Validating the WLS_MFT1 Managed Server on MFTHOST1	18-39
Configuring Web Services Manager	18-40
Updating WebServices Domain Configuration	18-40
Bootstrapping WSM	18-40
Propagating the Domain and Starting the Servers on MFTHOST2	18-42
Unpacking the Domain Configuration on MFTHOST2	18-42
Starting the Node Manager on MFTHOST2	18-43
Starting and Validating the WLS_MFT2 Managed Server on MFTHOST2	18-43
Modifying the Upload and Stage Directories to an Absolute Path	18-43
Configuring the Web Tier for the Extended Domain	18-43
Setting Frontend Addresses and WebLogic Plugin for the MFT Cluster and the Administration Server	18-43
Configuring Oracle HTTP Server for Managed File Transfer	18-44
Validating the Managed File Transfer URLs Through the Load Balancer	18-46
Configuring and Enabling the SSH-FTP Service for Managed File Transfer	18-47
Generating the Required SSH Keys	18-47
Configuring the SFTP Ports	18-48
Configuring Oracle Load Balancer for SFTP Services	18-49
Additional SFTP Configuration Steps for Managed File Transfer	18-49
Creating a New LDAP Authenticator and Provisioning Users for Managed File Transfer	18-50
Replacing Connect Strings with the Appropriate TNS Alias	18-51
Backing Up the Configuration	18-51

Part IV Common Configuration and Management Procedures for an Enterprise Deployment

19 Common Configuration and Management Tasks for an Enterprise Deployment

Configuration and Management Tasks for All Enterprise Deployments	19-1
Verifying Appropriate Sizing and Configuration for the WLSRuntimeSchemaDataSource	19-2
Verifying Manual Failover of the Administration Server	19-3

Failing Over the Administration Server When Using a Per Host Node Manager	19-4
Validating Access to the Administration Server on SOAHOST2 Through Load Balancer	19-5
Failing the Administration Server Back to SOAHOST1 When Using a Per Host Node Manager	19-5
Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment	19-6
Setting the Front End Host and Port for a WebLogic Cluster	19-7
About Using Third Party SSL Certificates in the WebLogic and Oracle HTTP Servers	19-8
Using Third Party SSL Certificates in WebLogic Servers	19-8
Using Third Party SSL Certificates in Oracle HTTP Servers	19-10
Enabling SSL Communication Between the Middle Tier and SSL Endpoints	19-12
When is SSL Communication Between the Middle Tier and the Frontend Load Balancer Necessary?	19-12
Generating Certificates, Identity Store, and Truststores	19-13
Importing Other External Certificates into the Truststore	19-13
Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts	19-14
Configuring Roles for Administration of an Enterprise Deployment	19-14
Summary of Products with Specific Administration Roles	19-14
Summary of Oracle SOA Suite Products with Specific Administration Groups	19-15
Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group	19-15
Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group	19-16
Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment	19-17
Products and Components that use JMS Persistence Stores and TLOGs	19-17
JDBC Persistent Stores vs. File Persistent Stores	19-18
Using JDBC Persistent Stores for TLOGs and JMS in an Enterprise Deployment	19-20
About JDBC Persistent Stores for Web Services	19-26
Best Configuration Practices When Using RAC and Gridlink Datasources	19-26
Using TNS Alias in Connect Strings	19-27
Performing Backups and Recoveries for an Enterprise Deployment	19-31
Online Domain Run-Time Artifacts Backup/Recovery Example	19-32
Configuration and Management Tasks for an Oracle SOA Suite Enterprise Deployment	19-38
Deploying Oracle SOA Suite Composite Applications to an Enterprise Deployment	19-39
Using Shared Storage for Deployment Plans and SOA Infrastructure Applications Updates	19-39
Managing Database Growth in an Oracle SOA Suite Enterprise Deployment	19-39
Managing the JMS Messages in a SOA Server	19-40
Draining the JMS Messages from a SOA Server	19-40
Importing the JMS Messages into a SOA Server	19-43

20 Using Service Migration in an Enterprise Deployment

About Automatic Service Migration in an Enterprise Deployment	20-1
Understanding the Difference between Whole Server and Service Migration	20-1
Implications of Service Migration in an Enterprise Deployment	20-2
Understanding Which Products and Components Require Service Migration	20-3
Creating a GridLink Data Source for Leasing	20-3
Configuring Automatic Service Migration in an Enterprise Deployment	20-5
Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster	20-5
Configuring Automatic Service Migration	20-6
Changing the JTA Migration Settings for the Managed Servers in the Cluster	20-6
About Selecting a Service Migration Policy	20-6
Setting the Service Migration Policy for Each Managed Server in the Cluster	20-7
Validating Automatic Service Migration	20-7
Failing Back Services After Automatic Service Migration	20-9

21 Scaling Procedures for an Enterprise Deployment

Scaling Out the Topology	21-1
Prerequisites for Scaling Out	21-1
Scaling Out a Cluster	21-1
Verifying the Scale Out	21-13
Scaling in the Topology	21-14
Scaling Up the Topology	21-16
Prerequisites for Scaling Up	21-17
Scaling Up	21-17
Verifying the Scale Up of Clusters	21-26
Scaling Down the Topology	21-27

A Targeting Applications and Resources to Servers

Oracle SOA Enterprise Application Targets	A-1
Oracle SOA Enterprise Deployment Library Targets	A-4
Oracle SOA Enterprise Deployment Startup Class Targets	A-10
Oracle SOA Enterprise Deployment Shutdown Class Targets	A-11
Oracle SOA Enterprise Deployment JMS System Resource Targets	A-11
Oracle SOA Enterprise Deployment JDBC System Resource Targets	A-12

Preface

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

- [Audience](#)
- [Documentation Accessibility](#)
- [Diversity and Inclusion](#)
- [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>.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

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

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.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Understanding an Enterprise Deployment

It is important to understand the concept and general characteristics of a typical enterprise deployment, before you configure the Oracle SOA Suite 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.
- [About the Oracle SOA Suite Enterprise Deployment Topology](#)

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. Enterprise Deployment Guides are the foundation to Maximum Availability Architectures for Oracle FMW products. Oracle's Maximum Availability Architecture provides a set of architectures, configurations and operational best practices that provide High Availability and Disaster Recovery solutions for the entire Oracle Stack. An Enterprise Deployment is the incarnation of these best practices in the scope of a single data center. When combined with the appropriate configuration and operational models for Disaster Protection, the Enterprise Deployment will achieve optimal high availability, data protection and disaster recovery at the lowest cost and complexity, while providing the best Recovery Time Objective (RTO) and Recovery Point Objective (RPO).
- [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. Enterprise Deployment Guides are the foundation to Maximum Availability Architectures for Oracle FMW products. Oracle's Maximum Availability Architecture provides a set of architectures, configurations and operational best practices that provide High Availability and Disaster Recovery solutions for the entire Oracle Stack. An Enterprise Deployment is the incarnation of these best practices in the scope of a single data center. When combined with the appropriate configuration and operational models for Disaster Protection, the Enterprise Deployment will achieve optimal high availability, data protection and disaster recovery at the lowest cost and complexity, while providing the best Recovery Time Objective (RTO) and Recovery Point Objective (RPO).

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:

- To install a **development environment**, use the instructions in Installing Oracle SOA Suite Quick Start for Developers in *Installing SOA Suite and Business Process Management Suite Quick Start for Developers*.

A development environment provides the software and tools that you can use to develop Java, Oracle Application Development Framework, and other applications that depend on Oracle technologies. Development environments are typically installed on a single host and do not require many of the features of a production environment.

- 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.

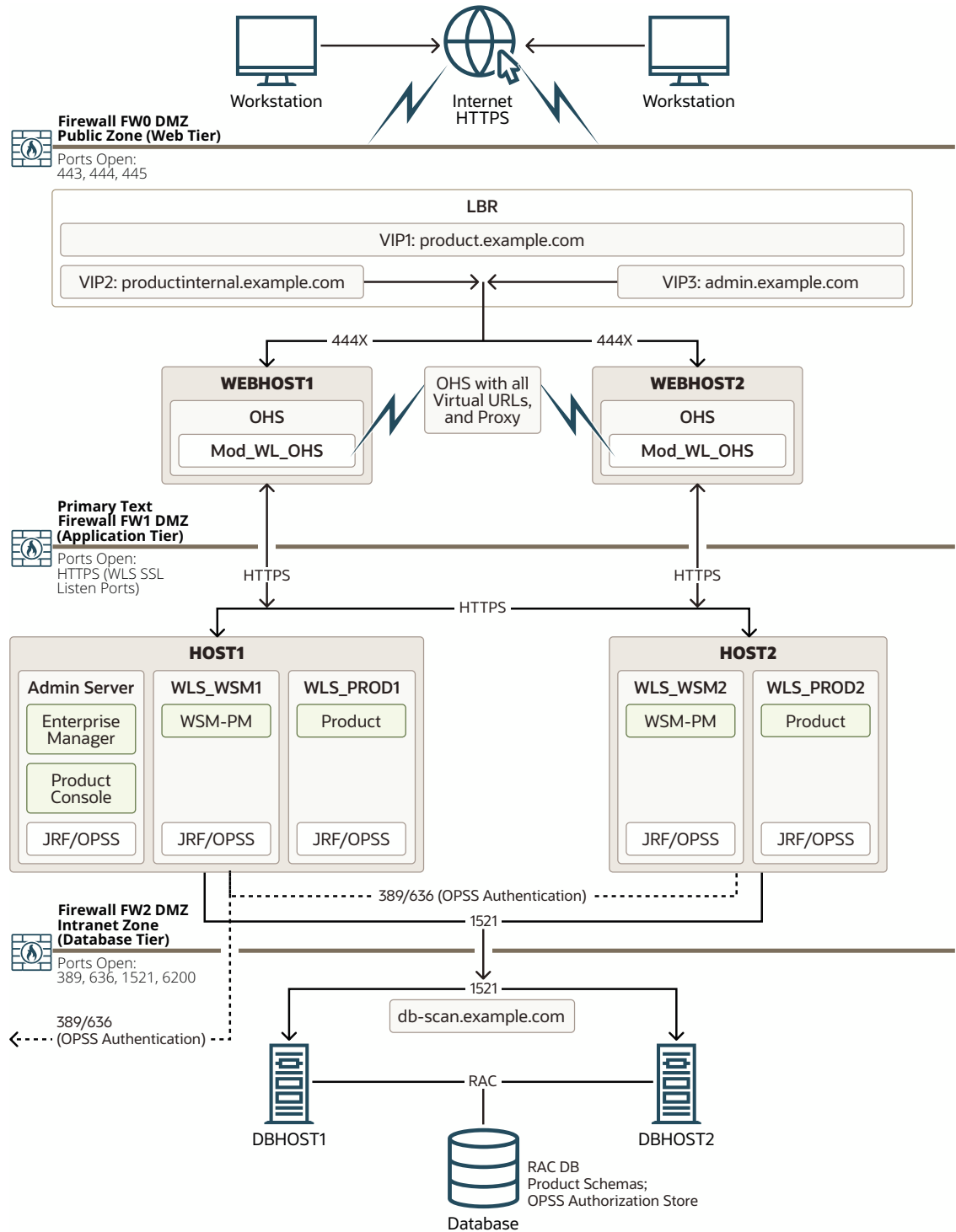
A best practices approach starts with the basic concept of a multi-tiered deployment and standard communications between the different software tiers.

This Enterprise Deployment uses secured communications using SSL all the way from the external clients to the backend WebLogic Servers. Although this eliminates numerous security vulnerabilities and increases the resiliency against different types of attacks, it has implications on the overall performance of the system. These implications vary depending on the applications deployed and the workloads in each invocation. For more information, see *Configuring SSL in Administering Security for Oracle WebLogic Server*. An alternative approach, is to terminate SSL at the load balancer. This approach also ensures the communication between the client and the load balancer, whilst maximizing performance within the system components by avoiding SSL overhead in the rest of the tiers. The downside is that this may not offer sufficient security in cloud based applications.

[Typical Enterprise Deployment Topology Diagram](#) 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 the virtual server names that are 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.
- 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 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. 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. This SSL termination at the load balancer provides a performance gain to applications at the cost of communications being secured in the rest of the tiers only by subnet restrictions in corporate networks. Given the increased security requirements applying nowadays to production deployments, this version of the Enterprise Deployment guide is implementing end-to-end SSL communication all the way from

the external clients to the middle tier. HTTPS protocol is used for the communication between clients and the front-end load balancer, between the frontend load balancer and Oracle HTTP Servers and between Oracle HTTP Server and the WebLogic Servers . This comes at the cost of performance overhead that is especially relevant in SSL handshakes. Depending on the type of connections used by client requests (long lived, short lived, keep alive settings) this may be a factor in the performance of an Enterprise Deployment. It is considered however that security standards have increased nowadays and this guide will focus on the most secure deployment approach.

 **Note:**

For the purpose of providing the most realistic working environment possible this guide uses demo certificates in the web and application tiers. Notice, however, that demonstration digital certificates and keystores are not recommended in production mode. The sample steps provided in this book should be executed or substituted with appropriate certificates signed by a formal Certificate Authority in your production environment.

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](#)
- [Load Balancer Considerations for Disaster Recovery and Multi-Data Center Topologies](#)
- [SFTP Requests for Oracle MFT Integration](#)
- [Specific Internal-Only Communications Between the Components of the Application Tier](#)

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](#).

HTTPS/encrypted requests are routed from the load balancer to the web tier. This guide provides instructions for SSL configuration between the load balancer and the web tier, and 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.

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 is 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.

SFTP Requests for Oracle MFT Integration

When MFT is deployed, the load balancer also needs to configure a TCP Virtual Server that will load balance the sFTP requests. The sFTP protocol is the secure protocol that is used to provide file transfers for MFT in the Enterprise Deployment Guides. See Embedded FTP and sFTP Servers in *Using Oracle Managed File Transfer*.

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.

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 [Typical Enterprise Deployment Topology Diagram](#), the following virtual server names are recognized by the hardware load balancer in this topology:

- `product.example.com`: This virtual server name is used for all incoming traffic.
Users enter this URL to access the Oracle Fusion Middleware product 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 Remote 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

This Enterprise Deployment Guide uses SSL for all the virtual servers, hence the frontend port 80 is no longer used. It is a best practice to assign the main external URL (for example, `https://myapplication.example.com`) to the SSL port number 443.



Note:

If port 80 remains open in the load balancer, then it is recommended to redirect any requests to it (non-SSL protocol) to port 443 (SSL protocol). Refer to your load balancer's specific documentation to implement this redirection. 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 are configured in the web tier. The following topics provide more detail.

- [Benefits of Using a Web Tier to Route Requests](#)
- [Alternatives to Using a Web Tier](#)

- [Configuration of Oracle HTTP Server in the Web Tier](#)
- [About Mod_WL_OHS](#)

Benefits of Using a Web Tier to Route Requests

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

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

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

Alternatives to Using a Web Tier

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

This approach provide the following advantages:

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

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

Configuration of Oracle HTTP Server in the Web Tier

This enterprise deployment guide provides information about configuring the Oracle HTTP Server instances as separate standalone domains, one on each web tier host. In each OHS Host the corresponding Node Manager listens only on localhost and manages only its local HTTP Server. See [Configuring Oracle HTTP Server for an Enterprise Deployment](#). Oracle recommends this approach instead of configuring Oracle HTTP Server instances as part of the application tier domain.

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](#)
- [About Coherence Clusters In a Typical Enterprise Deployment](#)

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. With the EDG topology model you can scale out or scale up the SOA components separately from WSMPM and the other way around. The same applies to patching procedures where downtime and restarts caused when applying patches have lower

impact when Oracle Web Services Manager servers are isolated from the rest of the components.

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 the services that are targeted for each cluster.

These best practices consider 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.

For another example, in an Oracle SOA Suite deployment, you might deploy Oracle SOA Suite and Oracle Service Bus on different hosts. Similarly, you might target Oracle Business Activity Monitoring and Enterprise Scheduler to a separate cluster on separate host computers.

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. Perform the appropriate resource planning and sizing, based on the system requirements for each type of server and the load that the system must sustain. Based on these decisions, you must adapt the steps to install and configure these variations accordingly from the instructions presented in this guide.

The Enterprise deployment guide is focused on a static cluster-based topology. Static clusters, also known as "configured clusters" are conventional clusters where you manually configure and add each server instance. Dynamic clusters (consisting of server instances that can be dynamically scaled up to meet the resource needs of your application) are out of the scope of this guide.

Mixed clusters (clusters that contains both dynamic and configured server instances) are not supported in a SOA enterprise deployment.

About the Node Manager Configuration in a Typical Enterprise Deployment

Oracle WebLogic 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, Oracle FMW products such as Oracle SOA Suite 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 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.

For scalability and manageability reasons, this Enterprise Deployment Guide uses Per Host Node manager configuration. The sections in the different chapters will provide the required steps for using a single Node manager in each host of the topology.

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 Coherence Clusters In a Typical Enterprise Deployment

The standard Oracle Fusion Middleware enterprise deployment includes a Coherence cluster that contains storage-enabled Managed Coherence Servers. Oracle FMW products add their clusters as members to this default coherence cluster during domain creation or extension.

This configuration is a good starting point for using Coherence. Depending upon your specific requirements, you can consider tuning and reconfiguring Coherence to improve performance in a production environment

Note:

Most Oracle Fusion Middleware products include Coherence GAR deployments. These deployments may have specific requirements pertaining to the default Coherence Cluster configuration (for example, local caches versus distributed). Consult the appropriate product installation and administration guides for specific limitations or processes regarding Coherence cluster configuration changes.

When reviewing port assignments, note that the Oracle Fusion Middleware products and components default to a Well Known Address (WKA) list that uses the port specified on the Coherence Clusters screen of the Configuration Wizard. The WKA list also uses the listen address of all servers that participate in the coherence cluster as the listen address for the WKA list. These settings can be customized by using the WLS Administration Console.

With respect to listen addresses, a Coherence cluster uses different services and protocols for network communications. The following are the out of the box services and their bind points:

- **Discovery Service** - Responsible for discovering other services including the cluster, defaults to a wildcard address, that is, listens on all addresses. It is configurable via operational configuration `coherence/cluster-config/unicast-listener/discovery-address` (generally left unset).
- **Clustering/TCMP** - Responsible for intra-cluster communication, defaults to whatever local address is routable to the WKA list, which are SOAHOST1 and SOAHOST2 ips in an enterprise deployment topology. It is configurable via operational configuration `coherence/cluster-config/unicast-listener/address` (generally left unset).
- **Extend Proxy** - Responsible for communication with non-clustered clients, defaults to the discovery address. It is configurable via `cache-cache-config/caching-schemes/proxy-scheme/acceptor-config/tcp-acceptor/local-address` (generally left unset).

For more information, refer to the following resources:

- For information about Coherence clusters, see Configuring and Managing Coherence Clusters in *Administering Clusters for Oracle WebLogic Server*.
- For information about tuning Coherence, see Performance Tuning in *Administering Oracle Coherence*.
- For information about storing HTTP session data in Coherence, see Using Coherence*Web with WebLogic Server in *Administering HTTP Session Management with Oracle Coherence*Web*.
- For more information about creating and deploying Coherence applications, see Creating Coherence Applications for WebLogic Server and Deploying Coherence Applications for WebLogic Server in *Developing Oracle Coherence Applications for Oracle WebLogic Server*.
- For information about the coherence listen addresses, see Element Reference and Configuring Caches in *Developing Applications with Oracle Coherence*.

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 FMW Products and the Oracle Platform Security Services (OPSS) policy store.

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

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

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

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

3

About the Oracle SOA Suite Enterprise Deployment Topology

The Oracle SOA Suite enterprise deployment topologies represent specific reference implementations of the concepts that are described in [About a Typical Enterprise Deployment](#).

- [About the Primary and Build-Your-Own Enterprise Deployment Topologies](#)
This guide focuses on one or more primary reference topologies for a selected product. In addition, this guide provides high-level information about how to design and build your own enterprise deployment topology.
- [Diagrams of the Primary Oracle SOA Suite Enterprise Topologies](#)
The two primary Oracle SOA Suite enterprise deployment topologies are: Oracle SOA Suite and Oracle Service Bus Topology and Oracle SOA Suite and Oracle Business Activity Monitoring Topology.
- [About the Primary Oracle SOA Suite Topology Diagrams](#)
Most of the elements of Oracle SOA Suite topologies represent standard features of any enterprise topology that follows the Oracle-recommended best practices. These elements are unique to the primary topology.
- [Flow Charts and Road Maps for Implementing the Primary Oracle SOA Suite Enterprise Topologies](#)
Instructions in the form of flow charts and road maps help you to install and configure the enterprise deployment topology with ease.
- [Building Your Own Oracle SOA Suite Enterprise Topology](#)
You can implement alternative topologies depending on the requirements of your organization, by using some variations of the instructions provided in this guide.
- [About Installing and Configuring a Custom Enterprise Topology](#)
If you choose to implement a topology that is not described in this guide, be sure to review the certification information, system requirements, and interoperability requirements for the products that you want to include in the topology.
- [About Using SSL Certificates in the Oracle SOA Suite Enterprise Topology](#)
As explained in previous chapters, the Oracle SOA Suite Enterprise Deployment Topology uses SSL all the way from the external clients to the backend WebLogic Servers. The HTTPS protocol is used for the communication between the clients and the front-end load balancer, the front-end load balancer and Oracle HTTP Servers and the Oracle HTTP Servers and the WebLogic Servers.
- [About Using JDBC Persistent Stores](#)
Oracle recommends that you use JDBC stores. This leverages the consistency, data protection, and high availability features of an Oracle database and makes stores available for all the servers in the cluster.
- [About Using Automatic Service Migration for the Oracle SOA Suite Enterprise Topology](#)
To ensure high availability of the Oracle SOA Suite products and components, this guide recommends that you enable Oracle WebLogic Server Automatic Service Migration for the clusters that you create as part of the reference topology.

- [About Reference Configuration for SOA and OSB](#)
During the installation process, you can create either a Reference Configuration domain or a Classic domain using the Templates screen of the Configuration Wizard. A Reference Configuration domain guards servers from running into out-of-memory, stuck threads, endpoint connectivity, and database issues.

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

This guide focuses on one or more primary reference topologies for a selected product. In addition, this guide provides high-level information about how to design and build your own enterprise deployment topology.

The exact topology you install and configure for your organization might vary, but for the primary topologies, this guide provides step-by-step instructions for installing and configuring those topologies.

For the build-your-own topologies, the guide also provides information about how to add specific components or products required for your specific environment.

Diagrams of the Primary Oracle SOA Suite Enterprise Topologies

The two primary Oracle SOA Suite enterprise deployment topologies are: Oracle SOA Suite and Oracle Service Bus Topology and Oracle SOA Suite and Oracle Business Activity Monitoring Topology.

- [Diagram of the Oracle SOA Suite and Oracle Service Bus Topology](#)
- [Diagram of the Oracle SOA Suite and Oracle Business Activity Monitoring Topology](#)

Diagram of the Oracle SOA Suite and Oracle Service Bus Topology

[Figure 3-1](#) shows a diagram of the Oracle SOA and Oracle Service Bus enterprise deployment topology.

 **Note:**

You can configure Oracle Service Bus in the same domain as Oracle SOA Suite or in its own domain. See [About the Topology Options for Oracle Service Bus](#).

For a description of the standard elements shown in the diagram, see [Understanding the Typical Enterprise Deployment Topology Diagram](#).

For a description of the elements shown in the diagram, see [Understanding the Primary Oracle SOA Suite Topology Diagrams](#).

Figure 3-1 Oracle SOA Suite and Oracle Service Bus Enterprise Deployment Reference Topology Diagram

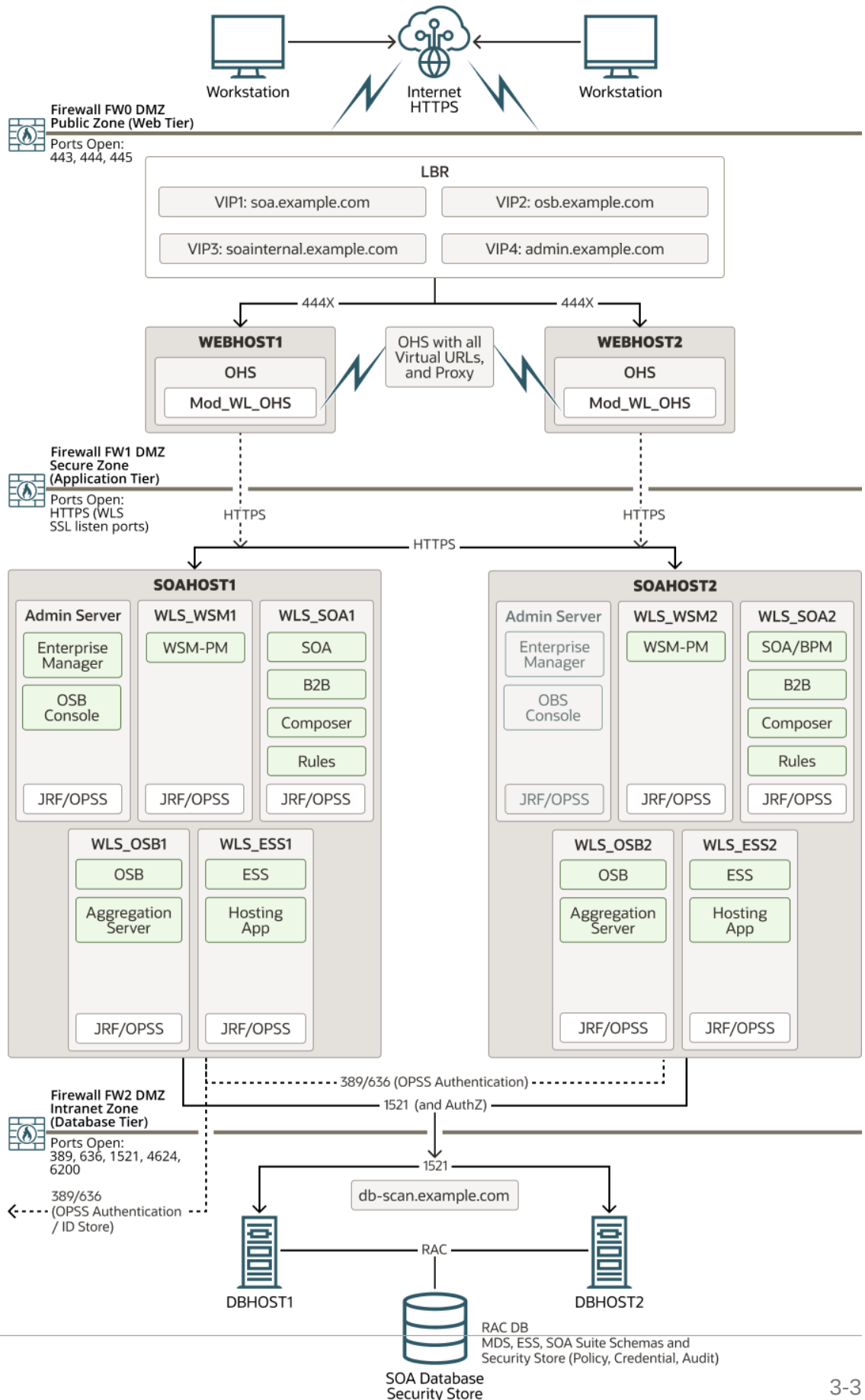


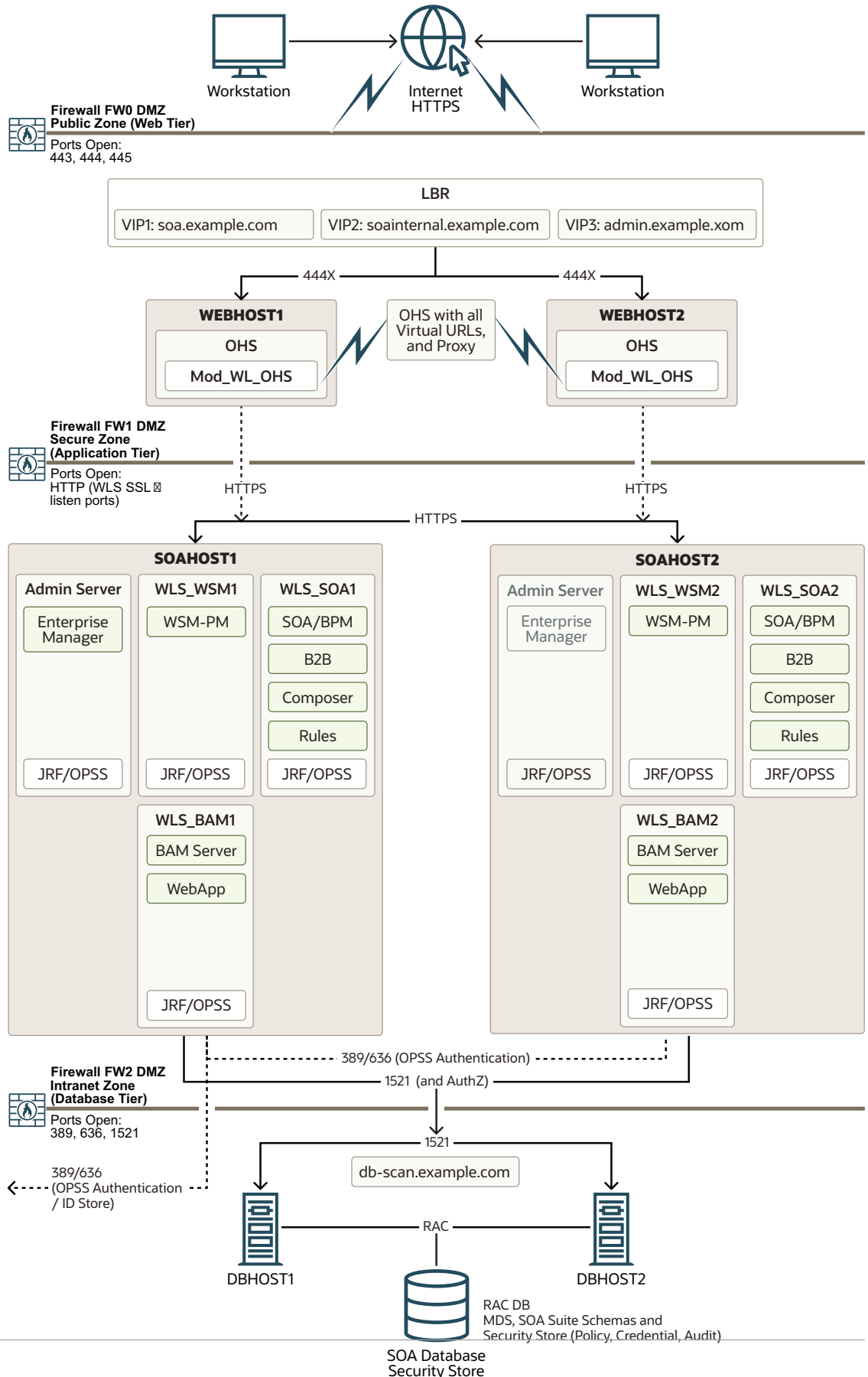
Diagram of the Oracle SOA Suite and Oracle Business Activity Monitoring Topology

[Figure 3-2](#) shows a diagram of the Oracle SOA Suite and Oracle Business Activity Monitoring enterprise topology.

For a description of the standard elements shown in the diagram, see [Understanding the Typical Enterprise Deployment Topology Diagram](#).

For a description of the elements that are specific to the Oracle SOA Suite topologies, see [Understanding the Primary Oracle SOA Suite Topology Diagrams](#).

Figure 3-2 Oracle SOA Suite and Oracle Business Activity Monitoring Enterprise Topology Diagram



About the Primary Oracle SOA Suite Topology Diagrams

Most of the elements of Oracle SOA Suite topologies represent standard features of any enterprise topology that follows the Oracle-recommended best practices. These elements are unique to the primary topology.

These elements are described in detail in [Understanding a Typical Enterprise Deployment](#).

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

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

- [About the Topology Options for Oracle Service Bus](#)
- [Summary of Oracle SOA Suite Load Balancer Virtual Server Names](#)
- [About the Routing of SOA Composite Requests](#)
- [Summary of the Managed Servers and Clusters on SOA Application Tier](#)

About the Topology Options for Oracle Service Bus

The Oracle SOA Suite and Oracle Service Bus topology diagram in this guide assumes a single domain that contains both SOA Suite and Oracle Service Bus. However, it is often advantageous to configure Oracle Service Bus in its own domain.

For example, consider separate domains when you are using Oracle Service Bus on an enterprise scale. In this scenario, you can then use Oracle Service Bus to route to multiple SOA domains and other services.

On the other hand, if you are using Oracle Service Bus primarily for mediating and providing routing for SOA Suite composite applications, configure Oracle Service Bus in the same domain, but in separate clusters for optimum performance and scalability.

When considering these options, take into account patching and other life cycle maintenance operations. For example, Oracle SOA Suite and Oracle Service Bus sometimes have differing patching requirements. If the two products are in separate domains, it can be easier to patch one without affecting the other.

Summary of Oracle SOA Suite 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 SOA Suite topologies:

- `soa.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 SOA components. The load balancer routes all requests to this virtual server name over SSL. As a result, clients access this service by using the following secure address:

`soa.example.com:443`

- `osb.example.com` : This virtual server name that acts as the access point for all HTTP traffic to the runtime Oracle Service Bus resources and proxy services. The load balancer routes all requests to this virtual server name over SSL. As a result, clients access this service by using the following secure address:

`osb.example.com:443`

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

Specifically, for the Oracle SOA Suite enterprise topology, this URL is used for both Oracle SOA Suite and Oracle Service Bus internal communications.

The load balancer routes all requests to this virtual server name over SSL. As a result, clients access this service by using the following secure address:

`soainternal.example.com:444`

Note that this URL can also be set as the URL to be used for internal service invocations while modeling composites or at runtime with the appropriate Enterprise Manager MBeans. See [More About the soainternal Virtual Server 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 Remote Console.

As a result, clients access this service by using the following secure address:

`admin.example.com:445`

 **Note:**

There are some components that use specific TCP Virtual Servers in the front end LBR for non- HTTP access to the system. This is the case of Oracle MFT. These virtual servers may use the same host name and different port or may use a different host name. Using a different host name may be a likely option when network tools are used for controlling traffic (for example, prioritizing the type of traffic based on the destination addresses). However, you will require an additional host name address in the required DNS systems.

To provide an easier mapping between frontend hostnames and backend OHS listeners, this guide presents different listeners (443, 444 and 445) in the frontend load balancer that map to the 443, 444 and 445 listeners in OHS. This allows easier segregation of traffic rules and SLAs for each type of access (LBR listener on 443 for apps, 444 for internal access and 445 for WLS domain administration). Alternatively, the same port (443) can be used in the frontend load balancer for all the different frontend hostnames. However, that OHS cannot host more than one SSL virtual host on the same IP address and port so it will still require separate listeners for each virtual host. For more information about the OHS configuration for the different products, see the following chapters.

Instructions later in this guide explain how to:

- 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 the virtual host names and the correct host computers.

About the Routing of SOA Composite Requests

The following topics provide additional information on configuring the enterprise deployment for Oracle SOA Suite composite applications.

- [More About the soainternal Virtual Server Name](#)
- [About Web Services Optimizations for SOA Composite Applications](#)
- [About Accessing SOA Composite Applications through Oracle HTTP Server](#)
- [About Accessing Oracle SOA Suite Composite Applications Through the Load Balancer](#)

More About the soainternal Virtual Server Name

The `soainternal.example.com` virtual server name functions exactly the same as the `soa.example.com`, except that it is invoked by intranet clients and callbacks only. This topic provides additional details.

The `soainternal.example.com` virtual server name is not used explicitly during the installation and configuration of the enterprise deployment, but custom systems often expose services that should be consumed by internal-only clients. In those cases, for efficiency and security reasons, you should avoid using an external URL such as `soa.example.com`. Instead, you should use an address that cannot be invoked by Internet clients. SOA composite applications, in particular, can use this internal URL in their end points, either directly or through deployment plans.

When you use the `soainternal.example.com` address, there are implications for the front end address specified for the system. Web services optimizations (for example, direct RMI invocation instead of invocations that involve a full loopback to the load balancer endpoint) are triggered when the front end address for the cluster matches the invocation endpoint. For this reason, depending on the number and relevance of the expected internal invocations, consider setting the front end URL for the cluster and the `ServerURL` and `HTTPServerURL` properties to either the external or internal.

You can set the front end URL for a cluster when you create the cluster in the Configuration Wizard. You can also modify it later, by using the WebLogic Remote Console.

For more information about setting the `ServerURL` and `HTTPServerURL` properties, see [Configuring SOA Infrastructure Properties](#).

About Web Services Optimizations for SOA Composite Applications

When you configure internal callbacks so that SOA composite applications can communicate efficiently within the enterprise deployment, you should be aware of how the system checks for the proper end-point address for each request.

For webservice local optimization, the basic requirement is to make sure that the two SOA composites are colocated on the same Managed Server or process. To determine if the composites are colocated on the same server, Oracle SOA Suite compares the server on which the target service composite is deployed (host and port configuration) with those specified in the reference service endpoint URI.

- For target service host value, here is the sequence of checks in order of precedence:
 - Checks the Server URL configuration property value on SOA Infrastructure Common Properties page.

- If not specified, checks the FrontendHost and FrontendHTTPPort (or FrontendHTTPSPort if SSL is enabled) configuration property values from the cluster MBeans.
- If not specified, checks the FrontendHost and FrontendHTTPPort (or FrontendHTTPSPort if SSL is enabled) configuration property values from the Oracle WebLogic Server MBeans.
- If not specified, uses the DNS-resolved Inet address of localhost.
- For target service port value, here is the sequence of checks in order of precedence:
 - Checks the port configured in HttpServerURL on SOA Infrastructure Common Properties page.
 - If not specified, checks the port configured in Server URL on SOA Infrastructure Common Properties page.
 - If not specified, checks the FrontendHost and FrontendHTTPPort (or FrontendHTTPSPort if SSL is enabled) configuration property values from the cluster MBeans.
 - If not specified, checks the FrontendHost and FrontendHTTPPort (or FrontendHTTPSPort if SSL is enabled) configuration property values from the Oracle WebLogic Server MBean.
 - If not specified, SOA Suite assumes 80 for HTTP URLs and 443 for HTTPS URLs. Notice that all frontend settings use the SSL port (443) in this EDG so it is recommended that the target services explicitly set it if neither of the HttpServerURL on SOA Infrastructure Common Properties have configured it. Target services that do not specify a port explicitly per one of the indicated mechanisms above, will fail since this guide no longer exposes port 80.

About Accessing SOA Composite Applications through Oracle HTTP Server

When you route requests from the Oracle HTTP Server instances on the web tier to specific Oracle SOA Suite composite application URLs on the application, consider the following:

- In previous releases of Oracle Fusion Middleware, if a request to Oracle SOA Suite composite application was received by the Managed Server and the composite application was not yet loaded, Oracle HTTP Server generated an HTTP 503 (Service Unavailable) message.
- Since Oracle Fusion Middleware 12c, this behavior changed. If requests for a composite arrives before the composite is active, then the HTTP requests are put on hold until the required artifacts are available and the composite reaches the active state.

 **Note:**

Composites that include JCA bindings, EJB, and ADF binding cannot be lazy loaded and act similar to composites that are yet loaded.

This change in behavior allows you to route requests to composite applications that are not yet loaded during the startup of an Oracle SOA Suite Managed Server. However, the communication channel between the Oracle HTTP Server and Oracle WebLogic Server needs to account for the possibility of long delays in getting replies.

To address this issue, while you configure firewalls between Oracle HTTP Server and Oracle WebLogic Server, set the appropriate timeout to avoid shutting down of connections that are

waiting for a composite to be loaded. See [Configuring the Firewalls and Ports for an Enterprise Deployment](#).

Note that the Oracle HTTP Server instances route requests based on the availability of the Oracle WebLogic Server servers and not on the availability of any specific application. The instances continue to route the requests as long as the Oracle WebLogic Server is up and running.

About Accessing Oracle SOA Suite Composite Applications Through the Load Balancer

In the default configuration, the hardware load balancer routes all requests to the web tier, which then routes the requests to the appropriate resource in the application tier.

However, you can configure the hardware load balancer to route directly to Managed Servers on the application tier. This configuration has some benefits, especially in an Oracle SOA Suite enterprise deployment:

- Configuration and processing overhead is lower than when you use Oracle HTTP Server.
- It enables monitoring at the application level, because the load balancer can be configured to monitor specific URLs in each WLS Server (something that is not possible with Oracle HTTP Server).

If Oracle HTTP server directs an HTTP request for a composite to a Oracle SOA Suite Managed Server and the `soa-infra` application is not yet active, then the request fails. Therefore, you should always verify that the `soa-infra` application is active after you start, restart, or migrate a server.

There is at least one disadvantage to this approach. If requests are routed directly from the load balancer to the Managed Servers, then each request crosses two firewalls without any proxy or interception. This might a security issue, depending on the network security policies in your organization.

Summary of the Managed Servers and Clusters on SOA 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 Oracle SOA Suite domain consists of the clusters shown in [Table 3-1](#). These clusters function as active-active high availability configurations.

Table 3-1 Summary of the Clusters in the Oracle SOA Suite Enterprise Deployment Topology

Cluster	Managed Servers
Oracle SOA Suite, Oracle Business Process Management, and Oracle B2B Cluster	WLS_SOA1, WLS_SOA2
Oracle Web Services Manager Cluster	WLS_WSM1, WLS_WSM2
Oracle Service Bus Cluster	WLS_OSB1, WLS_OSB2
Oracle Enterprise Scheduler	WLS_ESS1, WLS_ESS2
Oracle Business Activity Monitoring Cluster	WLS_BAM1, WLS_BAM2

There are some clusters that run in their own domains, such as MFT. The cluster for MFT is shown in [Table 3-2](#)

Table 3-2 Summary of the Cluster in the Oracle SOA Suite Enterprise Deployment Topology with Their Own Domains

Cluster	Managed Servers
Oracle Managed File Transfer	WLS_MFT1, WLS_MFT2

Flow Charts and Road Maps for Implementing the Primary Oracle SOA Suite Enterprise Topologies

Instructions in the form of flow charts and road maps help you to install and configure the enterprise deployment topology with ease.

The following sections summarize the high-level steps that you must perform to install and configure the enterprise topology that is described in this chapter.

- [Flow Chart of the Steps to Install and Configure the Primary Oracle SOA Suite Enterprise Topologies](#)
- [Roadmap Table for Planning and Preparing for an Enterprise Deployment](#)
- [Roadmap Table for Configuring the Oracle SOA Suite and Oracle Service Bus Enterprise Topology](#)
- [Roadmap Table for Configuring the Oracle SOA Suite and Oracle Business Activity Monitoring Enterprise Topology](#)

Flow Chart of the Steps to Install and Configure the Primary Oracle SOA Suite Enterprise Topologies

[Figure 3-3](#) shows a flow chart of the steps required to install and configure the primary enterprise deployment topologies that is described in this chapter. The sections following the flow chart explain each step in the flow chart.

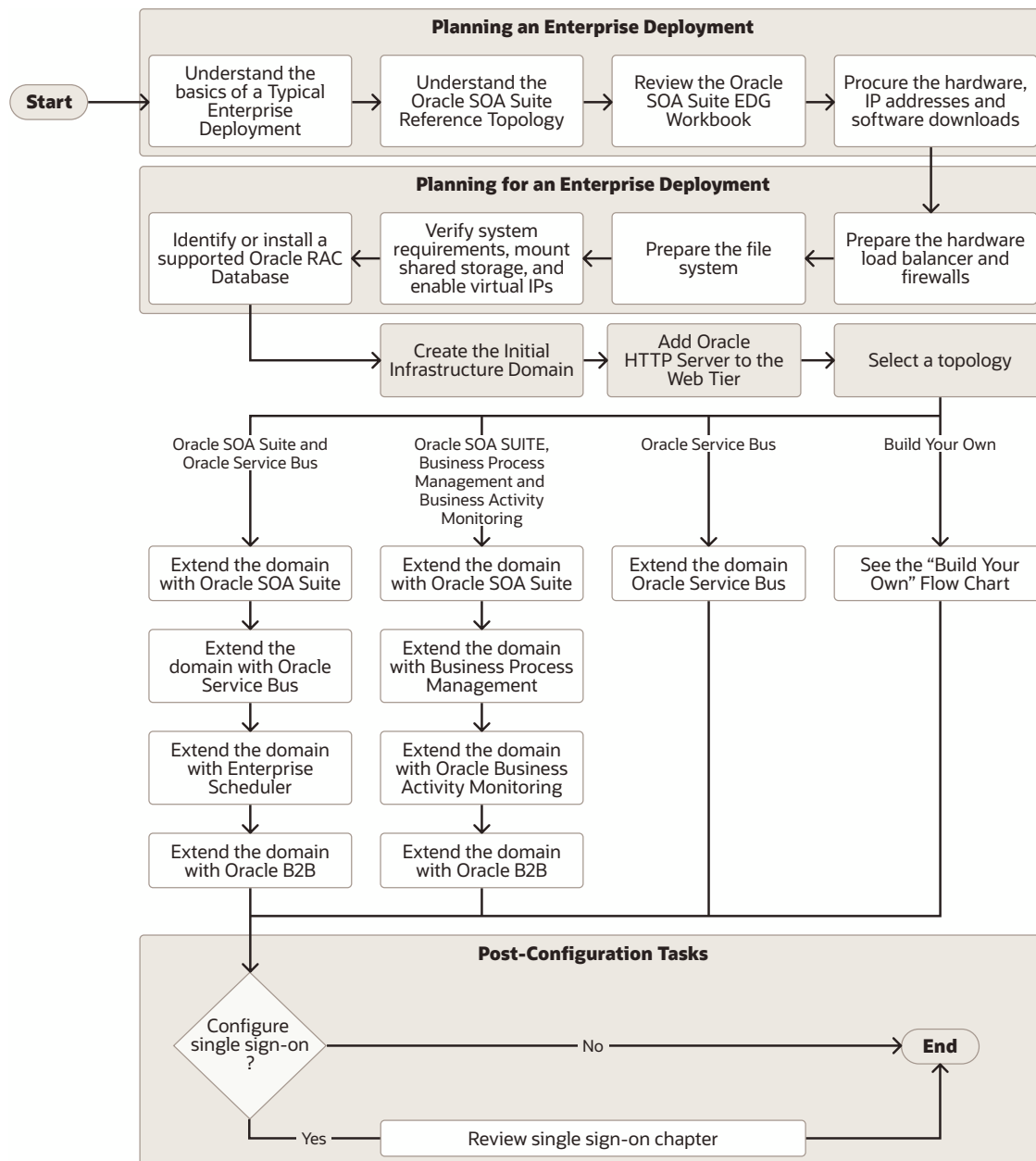
This guide is designed so you can start with a working Oracle SOA Suite domain and then later extend the domain to add additional capabilities.

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 SOA Suite 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 similar to 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 3-3 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

Flow Chart Step	More Information
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 SOA Suite 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 Oracle SOA Suite and Oracle Service Bus Enterprise Topology

[Table 3-3](#) describes each of the configuration steps that are required when you configure the Oracle SOA Suite and Oracle Service Bus topology shown in [Figure 3-1](#).

These steps correspond to the Oracle SOA Suite and Oracle Service Bus Topology steps shown in the flow chart in [Figure 3-3](#).



Note:

You can configure Oracle Service Bus in the same domain as Oracle SOA Suite or in its own domain. See [About the Topology Options for Oracle Service Bus](#).

Table 3-3 Roadmap Table for Configuring the Oracle SOA Suite and Oracle Service Bus Enterprise Topology

Flow Chart Step	More Information
Create the initial infrastructure domain	Creating the Initial Infrastructure Domain for an Enterprise Deployment
Extend the domain to include the web tier	Configuring the Web Tier for an Enterprise Deployment

Table 3-3 (Cont.) Roadmap Table for Configuring the Oracle SOA Suite and Oracle Service Bus Enterprise Topology

Flow Chart Step	More Information
Extend the domain with Oracle SOA Suite	Extending the Domain with Oracle SOA Suite
Extend the domain with Oracle Service Bus	Extending the Domain with Oracle Service Bus
Extend the domain with Enterprise Scheduler	Extending the Domain with Oracle Enterprise Scheduler Note that extending the domain with Enterprise Scheduler is optional; perform the procedure in this chapter only if you want to configure Enterprise Scheduler.
Extend the domain with Oracle B2B	Extending the Domain with Oracle B2B Note that extending the domain with Oracle B2B is optional; perform the procedures in this chapter only if you want to configure Oracle B2B.
Create a domain for Oracle Managed File Transfer	Configuring Oracle Managed File Transfer in an Enterprise Deployment Note that configuring a domain with Oracle Managed File Transfer is optional, you must perform the procedures in this chapter only, if you want to configure Oracle Managed File Transfer.

Roadmap Table for Configuring the Oracle SOA Suite and Oracle Business Activity Monitoring Enterprise Topology

[Table 3-4](#) describes each of the configuration steps that are required to configure the Oracle SOA Suite and Oracle Business Activity Monitoring topology shown in [Figure 3-2](#).

These steps correspond to the configuration steps shown for the Oracle SOA Suite Oracle Business Activity Monitoring topology in the flow chart in [Figure 3-3](#).

Table 3-4 Roadmap Table for Configuring the Oracle SOA Suite and Oracle Business Activity Monitoring Enterprise Topology

Flow Chart Step	More Information
Create the initial infrastructure domain	Creating the Initial Infrastructure Domain for an Enterprise Deployment
Extend the domain to include the web tier	Configuring the Web Tier for an Enterprise Deployment
Extend the domain with Oracle SOA Suite	Extending the Domain with Oracle SOA Suite
Extend the domain with Business Process Management	Extending the Domain with Business Process Management
Extend the domain with Oracle Business Activity Monitoring	Extending the Domain with Business Activity Monitoring
Extend the domain with Oracle B2B	Extending the Domain with Oracle B2B Note that extending the domain with Oracle B2B is optional; perform the procedures in this chapter only if you want to configure Oracle B2B.

Building Your Own Oracle SOA Suite Enterprise Topology

You can implement alternative topologies depending on the requirements of your organization, by using some variations of the instructions provided in this guide.

This document provides step-by-step instructions to configure the two primary enterprise topologies for Oracle SOA Suite, which are described in [Diagrams of the Primary Oracle SOA Suite Enterprise Topologies](#).

However, Oracle recognizes that the requirements of your organization may vary, depending on the specific set of Oracle Fusion Middleware products that you purchase and the specific types of applications that you deploy.

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

Note:

All managed servers of a component type in the domain must belong to that cluster. For example, Oracle Service Bus domains support only a single Service Bus cluster inside each domain.

- [Flow Chart of the Build Your Own Enterprise Topologies](#)
- [Description of the Supported Build Your Own Topologies](#)

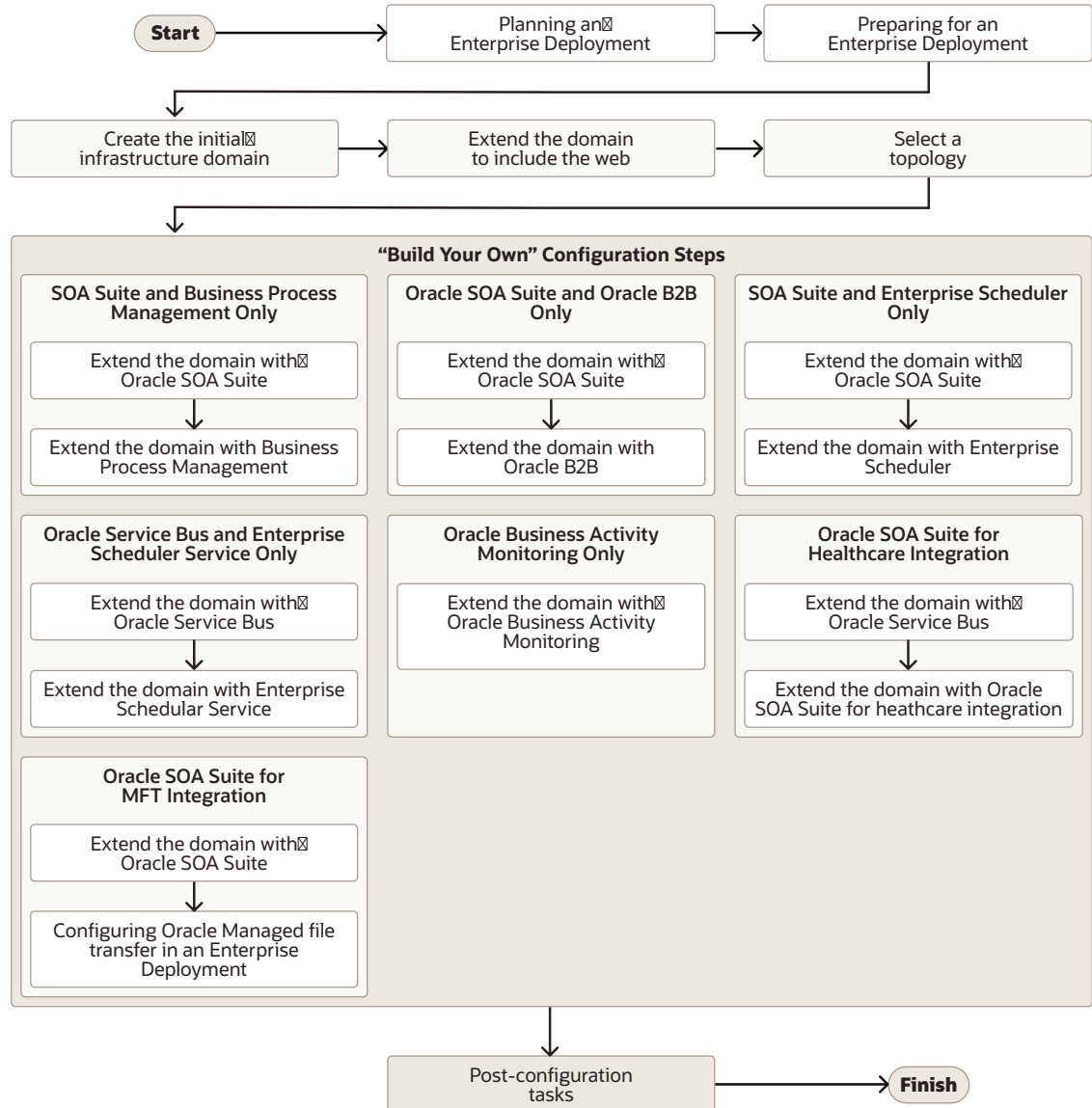
Flow Chart of the Build Your Own Enterprise Topologies

Building your own enterprise topology involves picking and choosing which Oracle Fusion Middleware products and which configuration steps you want to use to build your topology.

[Figure 3-4](#) shows the high-level configuration steps that are required to build some typical alternative Oracle SOA Suite enterprise topologies. Each of the configuration steps corresponds to a chapter in this guide.

Note that modifications of the steps in this guide are necessary in order to implement the Build Your Own topologies. Refer to [Description of the Supported "Build Your Own" Topologies](#) for more information.

Figure 3-4 Flow Chart of the Oracle SOA Suite Build-Your-Own Topologies



Description of the Supported Build Your Own Topologies

[Table 3-5](#) describes the configuration steps to follow if you want to use the instructions in this guide to build the enterprise topologies listed in [Figure 3-4](#).

It also identifies some differences you need to consider when you use the existing instructions in this guide to build each topology.

Table 3-5 Roadmap Table for Building Your Own Enterprise Topology

Topology	After You Configure the Web Tier, Refer to the Following Chapters	Considerations and Dependencies
SOA Suite and Business Process Management only	<ul style="list-style-type: none"> • Extending the Domain with Oracle SOA Suite • Extending the Domain with Business Process Management 	<p>These instructions assume you will run the Oracle SOA Suite and Business Process Management installer twice--once to install Oracle SOA Suite and once to install Oracle Business Process Management.</p> <p>Alternatively, you can install both Oracle SOA Suite and Oracle Business Process Management at the same time by selecting the BPM install type during the installation.</p> <p>Similarly, you can configure this topology by running the Configuration Wizard only once by selecting both the SOA and Oracle Business Process Management templates during the Configuration Wizard session.</p>
Oracle SOA Suite and Oracle B2B only	<ul style="list-style-type: none"> • Extending the Domain with Oracle SOA Suite • Extending the Domain with Oracle B2B 	No special instructions required.
SOA Suite and Enterprise Scheduler only	<ul style="list-style-type: none"> • Extending the Domain with Oracle SOA Suite • Extending the Domain with Oracle Enterprise Scheduler 	No special instructions required.
Oracle Service Bus and Enterprise Scheduler only	<p>See</p> <ul style="list-style-type: none"> • Extending the Domain with Oracle Service Bus • Extending the Domain with Oracle Enterprise Scheduler 	<p>This topology does not require Oracle SOA Suite. However, the instructions in Extending the Domain with Oracle Service Bus assume you have already created a cluster of two SOA Managed Servers.</p> <p>As a result, when you create this topology, ignore any references to the SOA Managed Servers or the SOA Cluster.</p> <p>In addition, you must run the Repository Creation Utility (RCU) to create the SOAINFRA schema, which is also required by Oracle Service Bus.</p>
Oracle Business Activity Monitoring only	Extending the Domain with Business Activity Monitoring	<p>The instructions in Extending the Domain with Business Activity Monitoring assume that you are extending an existing Oracle SOA Suite domain and that the Oracle SOA Suite software (which includes Oracle BAM) has already been installed in an Oracle home on shared storage.</p> <p>For this Oracle BAM-only topology, you need to install Oracle SOA Suite into the Oracle Fusion Middleware Infrastructure Oracle home before you can configure the domain to include an Oracle BAM cluster.</p> <p>In addition, you must run the Repository Creation Utility (RCU) to create the required SOA schemas.</p>
Oracle SOA Suite for MFT Integration	<ul style="list-style-type: none"> • Configuring Oracle Managed File Transfer in an Enterprise Deployment 	Oracle Managed File Transfer requires TCP listeners in the Load Balancer.

About Installing and Configuring a Custom Enterprise Topology

If you choose to implement a topology that is not described in this guide, be sure to review the certification information, system requirements, and interoperability requirements for the products that you want to include in the topology.

After you verify that the topology is supported, then you can either use the instructions in this guide as a guide to install and configure the components that you need, or you can install and configure a standard installation topology by using the Oracle Fusion Middleware 12c installation guides and use the *Start Small and Scale Out* approach to configure your environment.

For more information about planning your installation, see Planning for a Production Environment in *Planning an Installation of Oracle Fusion Middleware*.

About Using SSL Certificates in the Oracle SOA Suite Enterprise Topology

As explained in previous chapters, the Oracle SOA Suite Enterprise Deployment Topology uses SSL all the way from the external clients to the backend WebLogic Servers. The HTTPS protocol is used for the communication between the clients and the front-end load balancer, the front-end load balancer and Oracle HTTP Servers and the Oracle HTTP Servers and the WebLogic Servers.

For these communications to take place, each access point (whether the frontend load balancer, Oracle HTTP Server or Oracle WebLogic Server) needs to certify its identity and provide a public key to encrypt traffic. This is done using the appropriate SSL certificate in each access point. These SSL certificates are typically provided by formal certificate authorities (CAs). It is expected that in a real enterprise deployment, certificates are issued by one of these commercial entities and are present in the pertaining end points. Since each Certificate Authority has its own mechanisms to issue a SSL certificate, it is out of the scope of this Enterprise Deployment Guide to use and describe such processes. However, to provide a working end to end configuration with a solid encryption in the different communications across tiers, this guide provides SSL configuration steps using the WebLogic per domain Certificate Authority. The WebLogic per-domain CA is an enhanced Demo CA that uses the domain's secret key to encrypt passphrases and is automatically created with each WebLogic Domain. It adds an additional security level as compared to traditional self-signed certificates since it uses a strong and domain-specific key for encryption. This approach is considered secure and solid enough since the endpoints involved (OHS and WLS listeners) reside behind their corresponding firewalls and are not really exposed to the public. The front-end load balancer is considered a much more exposed endpoint and it is expected in all cases that users acquire the pertaining certificate for it from a formal Certificate Authority.

By default, the different chapters in this guide follow a flow based on the creation and usage of certificates using this per-domain CA. Scripts are provided to simplify most of the complexity around creating these certificates and adding them to the corresponding stores and wallets. This guide, also provides a section in the *Common Configuration and Management Tasks for an Enterprise Deployment* describing how to use a formal CA and its certificates. Refer to it if your system counts with formal certificates for all the Oracle HTTP Server and WebLogic Listeners.

About Using JDBC Persistent Stores

Oracle recommends that you use JDBC stores. This leverages the consistency, data protection, and high availability features of an Oracle database and makes stores available for all the servers in the cluster.

Refer to the technical details in the [Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment](#) section in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter. These technical details provide some guidance on possible performance implications and tuning recommendations.

Always make the following selections in the **High Availability Options** screen in the Configuration Wizard when creating and extending domains:

- Set JTA Transaction Log Persistence to JDBC TLog Store.
- Set JMS Server Persistence to JMS JDBC Store.

This will guarantee that JDBC persistent stores are used both for both JMS and TLOGS.

About Using Automatic Service Migration for the Oracle SOA Suite Enterprise Topology

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

Service Migration is configured using the Configuration Wizard HA Options screen. When you select the **Enable Automatic Service Migration** option in the Configuration Wizard HA Options screen, it configures migratable target definitions that are required for automatic service migration.

In the same screen, you use **JTA Transaction Log Persistence** and **JMS Server Persistence** options to configure them with JDBC stores automatically. Oracle recommends that you enable these options when you configure clusters in the SOA enterprise deployment.

For more information about service migration, see [Using Service Migration in an Enterprise Deployment](#).

About Reference Configuration for SOA and OSB

During the installation process, you can create either a Reference Configuration domain or a Classic domain using the Templates screen of the Configuration Wizard. A Reference Configuration domain guards servers from running into out-of-memory, stuck threads, endpoint connectivity, and database issues.

A Reference Configuration domain supports SOA, OSB, ESS, and B2B topologies. The templates in these products include Reference Configuration in their names, and are the default templates listed in the Configuration Wizard for these products.

 **Note:**

- A SOA Reference Configuration domain does not support BPM and BAM components. The Reference Configuration feature does not apply to MFT domains.
- There is no specific Reference Configuration template for ESS. However, ESS can be added to both a Reference Configuration domain and to a Classic domain.

A Reference Configuration domain provides tuned parameters out-of-the-box for newly created SOA projects. Tuned parameters include but are not limited to:

- Java Virtual Machine: heap size, HTTP timeouts
- WebLogic Server: JTA timeout, HTTP extended logging
- Database: distributed_lock_timeout, db_securefiles
- Product-Specific: SOA, Service Bus, Adapters - Work Manager configuration, payload size restriction, and so on

This guide uses the Reference Configuration templates for SOA, OSB, and B2B components to take advantage of the tuned configuration. These reference configuration templates are shown in the Config Wizards as:

- Oracle SOA Suite Reference Configuration - 14.1.2.0.0 [soa]
- Oracle Service Bus Reference Configuration - 14.1.2.0.0 [osb]
- Oracle B2B Reference Configuration - 14.1.2.0.0 [soa]

For information about Reference Configuration, see:

- Selecting the Configuration Template for Oracle SOA Suite in *Installing and Configuring Oracle SOA Suite and Business Process Management*.
- Configuring a Reference Configuration Domain in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.
- Developing SOA Projects in Reference Configuration Mode in *Developing SOA Applications with Oracle SOA Suite*.

 **Note:**

If you plan to extend the SOA domain to add BPM or BAM, the SOA domain must be created using the Classic templates. These templates which do not implement the Reference Configuration optimizations are named Classic templates. Classic templates are also shown in the Configuration Wizard as follows:

- Oracle SOA Suite - 14.1.2.0.0 [soa] - SOA Classic template is used to extend an existing Infra domain with SOA Classic. This template is used in this guide to extend the Infra domain for scenarios where BPM or BAM will be added to the domain.
- Oracle B2B - 14.1.2.0 [soa] - B2B Classic template is used to extend an existing Classic domain with B2B Classic. This template is used in this guide to extend the SOA domain with B2B when it is a Classic domain (for scenarios where BPM or BAM will be added to the domain).
- Oracle Service Bus - 14.1.2.0.0 [osb] - OSB Classic template is used to extend an existing domain with OSB Classic. This is used in this guide to extend the Infra or SOA Classic domain for scenarios where BPM or BAM will be added to the domain.

Subsequent extensions on a Classic SOA domain (for B2B or OSB) must be done with Classic templates and not with Reference Configuration templates. Other SOA components (BPM, BAM, ESS, and MFT) do not have Reference Configuration templates, so the default templates shown in the Configuration Wizard for these products can be used to extend the SOA Classic domain.

- `$ORACLE_HOME/soa/common/templates/wls/oracle.soa.classic.domain_template.jar` is a SOA Classic template used to create new SOA Classic domains. This is a domain template and not an extension template. This template has dependencies on the basic WebLogic Server domain template (`wls.jar`) and the Oracle SOA Suite template (`oracle.soa_template.jar`). This template should be used only to create a new domain, not to extend any domain.
- `$ORACLE_HOME/soa/common/templates/wls/oracle.soa.b2b.classic.domain_template.jar` is a B2B Classic template used to create a new B2B Classic domains.
- `$ORACLE_HOME/osb/common/templates/wls/oracle.osb.classic.domain_template.jar` is a OSB Classic template used to create a new OSB Classic domain.

Each chapter in this guide will indicate the template to use in each case.

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 SOA Suite 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 SOA Suite Enterprise Deployment Workbook](#)
Locating and understanding the Oracle SOA Suite 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 SOA Suite Enterprise Deployment Workbook

Locating and understanding the Oracle SOA Suite Enterprise Deployment workbook enables you to use it efficiently.

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

- [Locating the Oracle SOA Suite Enterprise Deployment Workbook](#)
- [Understanding the Contents of the Oracle SOA Suite Enterprise Deployment Workbook](#)

Locating the Oracle SOA Suite Enterprise Deployment Workbook

The Oracle SOA Suite 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 SOA Suite Enterprise Deployment Workbook

The following sections describe the contents of the Oracle SOA Suite 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.

Using the Hardware - Host Computers Tab

The Hardware - Host Computers tab lists the host computers that are required to install and configure the Oracle SOA Suite 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 SOAHOST1, you can then replace the SOAHOST1 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 SOAHOST1 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.

On the **Network-Virtual Hostnames** enter the hostnames you are using as listen addresses in the different nodes. Ideally and in preparation for Disaster Protection, these hostnames should be virtual and not attached to a precise physical host (even if they do not map to a floating IP).

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 SOA Suite 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 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.
- In this Enterprise Deployment Guide, SSL listeners are used for the Oracle HTTP Servers and the Oracle WebLogic Servers. The load balancer should hence be able to establish SSL communication with the back-end servers in its pools.
- SSL acceleration (this feature is highly recommended, but not required for the enterprise topology).
- The ability to route TCP/IP requests; this is a requirement for Managed File Transfer, which can use sFTP/FTP protocol.

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 SOA Suite 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 SOA Suite 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 SOAHOST1, 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](#).

Table 5-1 Typical Memory, File Descriptors, and Processes Required for an Enterprise Deployment

Managed Server, Utility, or Service	Approximate Top Memory (SOA Classic Domain)	Approximate Top Memory (SOA Reference Configuration Domain)	Number of File Descriptors	Operating System Processes and Tasks
Administration Server	3.5 GB	4 GB	3500	165
WLS_WSM	3.0 GB	8 GB	2000	130
WLS_SOA	4.0 GB	8 GB	3100	240
WLS_OSB	4.0 GB	8 GB	2200	180
WLS_ESS	3.5 GB	8 GB	1300	35
WLS_BAM	3.5 GB	N/A**	2300	210
WLS_MFT	2 GB	N/A**	2000	350
WLST (connection to the Node Manager)	1.5 GB	1.5 GB	910	20
Configuration Wizard	1.5 GB	1.5 GB	700	20
Node Manager	1.0 GB	1.0 GB	720	15
TOTAL	29.0 GB*	40 GB*	19000	1550

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

** not supported in Reference Configuration Domains.

You can create either a Reference Configuration domain or a Classic domain by using the Templates screen of the Configuration Wizard, during installation. A Reference Configuration domain guards servers from running into out-of-memory, stuck threads, endpoint connectivity, and database issues.

The Reference Configuration domain supports SOA, OSB, SOA + OSB, and B2B topologies and Oracle recommends it in the SOA Enterprise Deployment Guide for the components that support it.

In a Reference Configuration domain, the min and max heap memory (-Xms -Xmx) configured for each server are greater than in Classic Domains: they are set to 4 GB for the Admin Server and 8 GB for the managed servers.

In case you need to tune these parameters, they are specified in the file `$ORACLE_HOME/soa/common/bin/setSOARefConfigEnv.sh`

```
# JVM Memory Arguments
  if [ "${STARTUP_GROUP}" = "AdminServerStartupGroup" ] ; then
    MEM_ARGS_NEW_MIN="-Xms4g"
    export MEM_ARGS_NEW_MIN

    MEM_ARGS_NEW_MAX="-Xmx4g"
    export MEM_ARGS_NEW_MAX
  else
    MEM_ARGS_NEW_MIN="-Xms8g"
    export MEM_ARGS_NEW_MIN

    MEM_ARGS_NEW_MAX="-Xmx8g"
    export MEM_ARGS_NEW_MAX
  fi
```

See *Configuring a Reference Configuration Domain* in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

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 14c (14.1.2.0.0) products, including the Oracle SOA Suite 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 SOA Suite 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
SOAHOST n (SOA only)	10 GB*
SOAHOST n (SOA and OSB)	11 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 and a virtual host name mapped to this VIP
- VIPs are not required for any of the Managed Servers in the FMW SOA Enterprise Deployment since all components support Automatic Service Migration.

You can then work with your network administrator to be sure that the Administration Server's VIP is 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 VIP --and the virtual host name to which it is mapped-- is 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 SOA Suite 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 VIP --and the virtual host name to which it is mapped-- is reserved and enabled on the corporate network.

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

The SOA Suite products support Automatic Service Migration. As a result, it is no longer necessary to reserve VIPs for each of the Managed Servers in the domain. Instead, a VIP is required for the Administration Server only.

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 Admin Server. The reassignment of virtual IP address for the Administration Server must be performed manually.

Note:

Regardless the use of virtual or physical IPs, Oracle also recommends that you use aliases to map to different IPs in different data centers in preparation for disaster recovery. It is recommended to use these aliases to configure the listen address for the components. This approach will be used in this guide.

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 SOA Suite 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 5-2](#).

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 5-2 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.

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](#).

 **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.

Table 5-3 List of Oracle Fusion Middleware Distributions

Distribution	Description	Installer File Name
Oracle Fusion Middleware 14c (14.1.2.0.0) Infrastructure	Download this distribution to install the Oracle Fusion Middleware Infrastructure, which includes Oracle WebLogic Server and Java Required Files software required for Oracle Fusion Middleware products. This distribution also installs the Repository Creation Utility (RCU), which in previous Oracle Fusion Middleware releases was packaged in its own distribution.	fmw_14.1.2.0.0_infrastructure.jar

Table 5-3 (Cont.) List of Oracle Fusion Middleware Distributions

Distribution	Description	Installer File Name
Oracle HTTP Server 14c (14.1.2.0.0)	Download this distribution to install Oracle HTTP Server on the Web tier hosts.	fmw_14.1.2.0.0_ohs_linux64.bin
Oracle Fusion Middleware 14c (14.1.2.0.0) SOA Suite and Business Process Management	Download this distribution to install the SOA Foundation and BPM software, which includes Oracle Business Activity Monitoring (BAM) and Oracle Enterprise Scheduler (ESS).	fmw_14.1.2.0.0_soa.jar
Oracle Fusion Middleware 14c (14.1.2.0.0) Service Bus	Download this distribution if you plan to install and configure Oracle Service Bus as part of the Oracle SOA Suite enterprise topology.	fmw_14.1.2.0.0_osb.jar
Oracle Fusion Middleware 14c (14.1.2.0.0) B2B and Healthcare	Download this distribution if you plan to install and configure Oracle B2B or Oracle B2B Healthcare as part of the Oracle SOA Suite enterprise topology.	fmw_12.2.1.4.0_b2bhealthcare.jar
Oracle Fusion Middleware 14c (14.1.2.0.0) Managed File Transfer	Download this distribution if you plan to install and configure Oracle Managed File Transfer as part of the Oracle SOA Suite enterprise topology.	fmw_14.1.2.0.0_mft.jar

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 that are used by various Oracle Fusion Middleware products and services. This ensures that the same port number is not used by two services on the same host, and that the proper ports are open on the firewalls in the enterprise topology.

Configuring Virtual Hosts on the Hardware Load Balancer

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

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

- [Overview of the Hardware Load Balancer Configuration](#)
- [Typical Procedure for Configuring the Hardware Load Balancer](#)
- [Summary of the Virtual Servers Required for an Enterprise Deployment](#)
- [Additional Instructions for admin.example.com](#)
- [Additional Instructions for soa.example.com](#)
- [Additional Instructions for soainternal.example.com](#)
- [Additional Instructions for osb.example.com](#)
- [Additional Instructions for mft.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. At the same time, this monitoring should not overload the backend system with too frequent health requests. In the end, a trade off needs to be made between how fast the death detection occurs and how much overhead is introduced on the systems that are monitored

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 load balancing between the web hosts, create a pool of servers that would direct requests to hosts WEBHOST1 and WEBHOST2 to each port used in the OHS. For example, a pool to WEBHOST1 and WEBHOST2 to port 4443 for access to applications like SOA and OSB, another pool to WEBHOST1 and WEBHOST2 to port 4444 for internal accesses, and another pool to WEBHOST1 and WEBHOST2 to port 4445 for access to admin consoles.

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 the internal addresses are only resolvable from inside the network.
- b. Assign the pool of servers created in *Step 1* to the virtual server.
- c. Configure SSL for the virtual server.
- d. Configure SSL for the communication with the pool of servers.

Some load balancers may need to be provided with the backend's certificate (the SSL certificate used by the OHS listeners in the backend pool) to establish the appropriate

SSL communication. In that case you may need to add the OHS's CA certificate to the load balancer as a trusted certificate. Since this guide uses example certificates based on the WebLogic per-domain CA, you can add this after the domain is created.

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 SOA Suite enterprise topology:

Table 6-1 List of Virtual Servers

Virtual Host	Server Pool	Protocol	External
admin.example.com:445	WEBHOST1.example.com:4445 WEBHOST2.example.com:4445	HTTPS	No
soa.example.com:443	WEBHOST1.example.com:4443 WEBHOST2.example.com:4443	HTTPS	Yes
soainternal.example.com:444	WEBHOST1.example.com:4444 WEBHOST2.example.com:4444	HTTPS	No
osb.example.com:443	WEBHOST1.example.com:4446 WEBHOST2.example.com:4446	HTTPS	Yes
mft.example.com:7022	SOAHOST1.example.com:7022 SOAHOST2.example.com:7022	TCP (SFTP)	Yes
mft.example.com:443	WEBHOST1.example.com:4443 WEBHOST2.example.com:4443	HTTPS	Yes

 **Note:**

If SOA Suite and Oracle Managed File Transfer are deployed on the same host, then Managed File Transfer can share the HTTPS virtual servers that are used by SOA to access the Managed File Transfer console. However, a separate Managed File Transfer virtual server is required for TCP protocol (used to load balance SFTP requests).

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.

Additional Instructions for soa.example.com

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

- Use port 443. If port 80 is used for customer usability, then it is recommended to redirect any requests to it (non-SSL protocol) to port 443 (SSL protocol). Refer to your load balancer's specific documentation to implement this redirection.
- Specify ANY as the protocol (non-HTTP protocols are required for B2B).
- Enable address and port translation.
- Enable reset of connections when services and nodes are down.
- Create rules to filter out access to `/management` and `/em` on this virtual server.

These context strings direct requests to the Oracle WebLogic Remote Console and to the Oracle Enterprise Manager Fusion Middleware Control and must be used only when you access the system from `admin.example.com`.

Note:

Oracle recommends that you configure LBR for cookie-based persistence because session persistence is required for some web applications of SOA, such as BPM Worklist (`/integration/worklistapp`), SOA Composer (`/soa/composer`), BPM Composer (`/bpm/composer`), BPM Workspace (`/bpm/workspace`), and so on.

Additional Instructions for soainternal.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 nodes are down.
- As with the `soa.example.com`, create rules to filter out access to `/console` and `/em` on this virtual server.

Additional Instructions for osb.example.com

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

- Use port 443. If port 80 is used for customer usability, then it is recommended to redirect any requests to it (non-SSL protocol) to port 443 (SSL protocol). Refer to your load balancer's specific documentation to implement this redirection.

- Enable address and port translation.
- Enable reset of connections when services and nodes are down.
- Create rules to filter out access to `/management` and `/em` on this virtual server.

These context strings direct requests to the Oracle WebLogic Remote Console and to the Oracle Enterprise Manager Fusion Middleware Control and should be used only when you access the system from `admin.example.com`.

Additional Instructions for `mft.example.com`

The Managed File Transfer requires a TCP virtual server in the load balancer for the Secure File Transfer Protocol (SFTP), in addition to the virtual server for HTTPS.

In the Managed File Transfer scenario, the load balancer directly routes the SFTP requests to the SFTP embedded servers. These SFTP embedded servers are running on the Managed File Transfer Managed Servers. For consistency, the port used in the hardware load balancer and in the SFTP servers is 7022. The Oracle HTTP Servers are not used for the SFTP requests because they cannot manage the SFTP protocol.

The Managed File Transfer also uses a HTTPS virtual server to access the MFT console. In this virtual server, the load balancer routes the HTTPS requests to the Oracle HTTP Servers.

Configuring the Firewalls and Ports for an Enterprise Deployment

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

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



Note:


The TCP/IP port for B2B is a user-configured port and is not predefined. Similarly, the firewall ports depend on the definition of TCP/IP ports.

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.



N
o
t
e
:
Y
o
u
n
e
e
d
t
h
i
s
o
p
t
i
o
n
o
n
l
y
i
f
r
e
d
i
r
e
c
t
i
o
n
f
r
o
m
p

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
			o r t 8 0 t o p o r t 4 4 3 i s s u e d .		
Browser request	FW0	44x	HTTPS / Load Balancer	Inbound	Timeout depends on the size and type of HTML content.
Browser request	FW1	44x	HTTPS / Load Balancer	Outbound (for intranet clients)	Timeout depends on the size and type of HTML content.
Callbacks and Outbound invocations	FW1	44x	HTTPS / Load Balancer	Outbound	Timeout depends on the size and type of HTML content.
Load balancer to Oracle HTTP Server	n/a	444X	HTTPS	n/a	n/a
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
WebLogic Remote Console and Enterprise Manager Console	FW1	9002	HTTPS / Remote Console and Enterprise Manager t3s	Both	You should tune this timeout based on the type of access to the Remote console (whether you plan to use the Oracle WebLogic Remote Console from the application tier clients or clients external to the application tier).
Database access	FW2	1521	SQL*Net	Both	Timeout depends on database content and on the type of process model used for SOA.
Coherence for deployment	n/a	9991 Coherence requires the following connectivity between members: <ul style="list-style-type: none"> Port 9991 for both UDP and TCP for both multicast and unicast configurations. TCP port 7. Ephemereal ports 32768-60999 for both udp and tcp. 	n/a	n/a	n/a
Oracle Unified Directory access	FW2	389 636 (SSL)	LDAP or LDAP/ssl	Inbound	You should tune the directory server's parameters based on load balancer, and not the other way around.
Oracle Notification Server (ONS)	FW2	6200	ONS	Both	Required for Gridlink. An ONS server runs on each database server.

Table 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
MFT SFTP Requests	FW0, FW1	7022	SFTP/Embedded SFTP servers in WLS_MFTn	Inbound	Timeout depends on the size of the transferred files.
MFT HTTP Requests	FW1	7010	HTTPS/WLS_MFTn	Inbound	Timeout depends on the size and type of the HTML content.

*External clients can access SOA servers directly on RMI or JMS (for example, for JDeveloper deployments and for JMX monitoring), in which case FW0 might need to be open or not depending on the security model that you implement.

Type	Firewall	Port and Port Range	Protocol/ Application	Inbound / Outbound	Other Considerations and Timeout Guidelines
WSM-PM access	FW1	7010	HTTPS / WLS_WSM-PMn	Inbound	Set the timeout to 60 seconds.
SOA Server access	FW1*	7004	HTTPS / WLS_SOAn	Inbound	Timeout varies based on the type of process model used for SOA.
Oracle Service Bus Access	FW1	8003	HTTPS / WLS_OSBN	Inbound/ Outbound	Set the timeout to a short period (5-10 seconds).
BAM access	FW1	7006	HTTPS / WLS_BAMn	Inbound	Connections to BAM WebApps are kept open until the report/browser is closed, so set the timeout as high as the longest expected user session.
Oracle Enterprise Scheduler access	FW1	7008	HTTPS/ WLS_ESSn	Inbound	-

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 <i>High Availability Guide</i>	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 <code>ASERVER_HOME</code> variable.
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. The Oracle home should be in the same location on each node.

About the Recommended Directory Structure for an Enterprise Deployment

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

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

The diagrams are used to indicate:

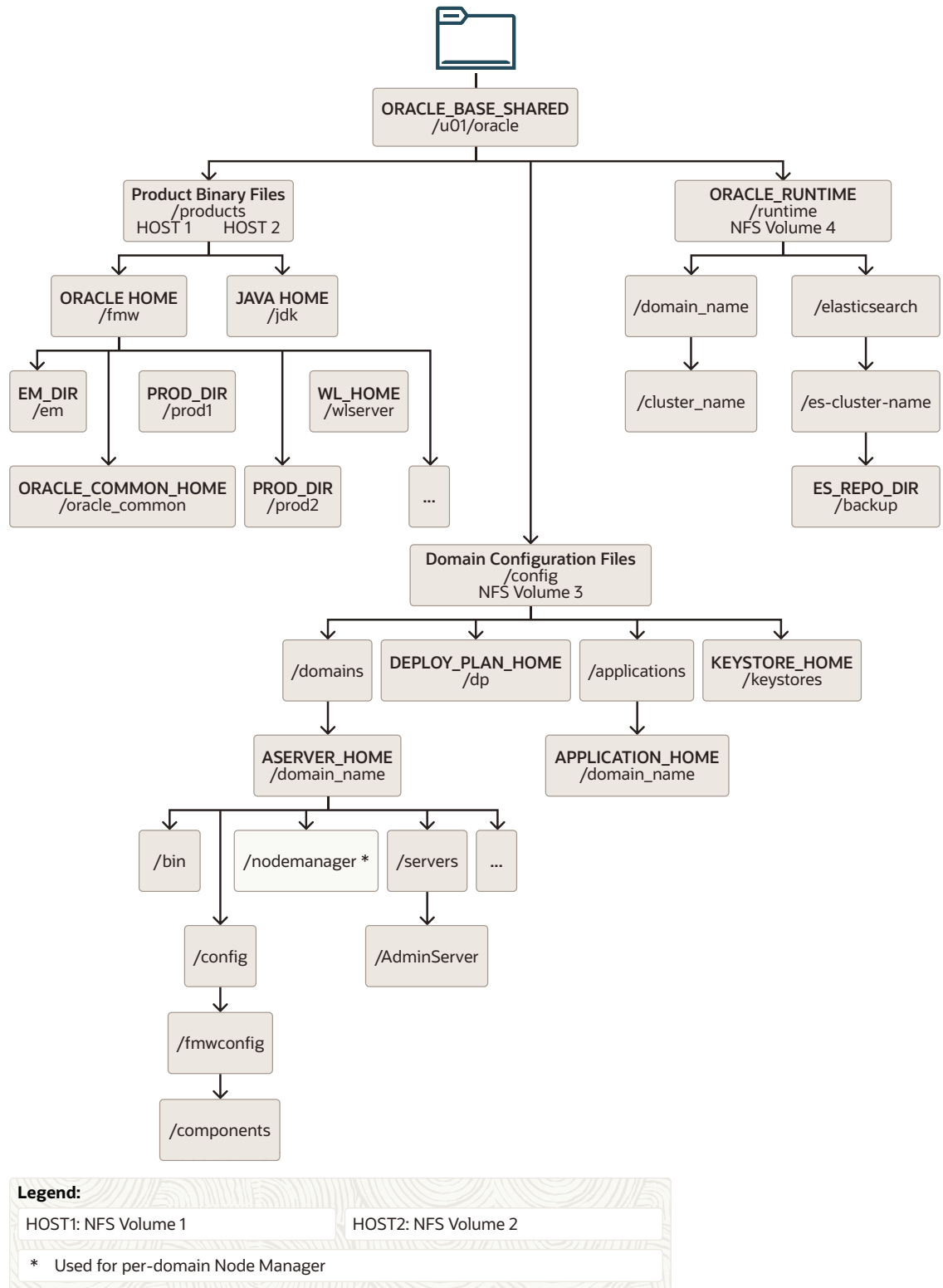
- [Figure 7-1](#) shows the resulting directory structure on the shared storage device after you have installed and configured a typical Oracle Fusion Middleware enterprise deployment. The shared storage directories are accessible by the application tier host computers.
- [Figure 7-2](#) shows the resulting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

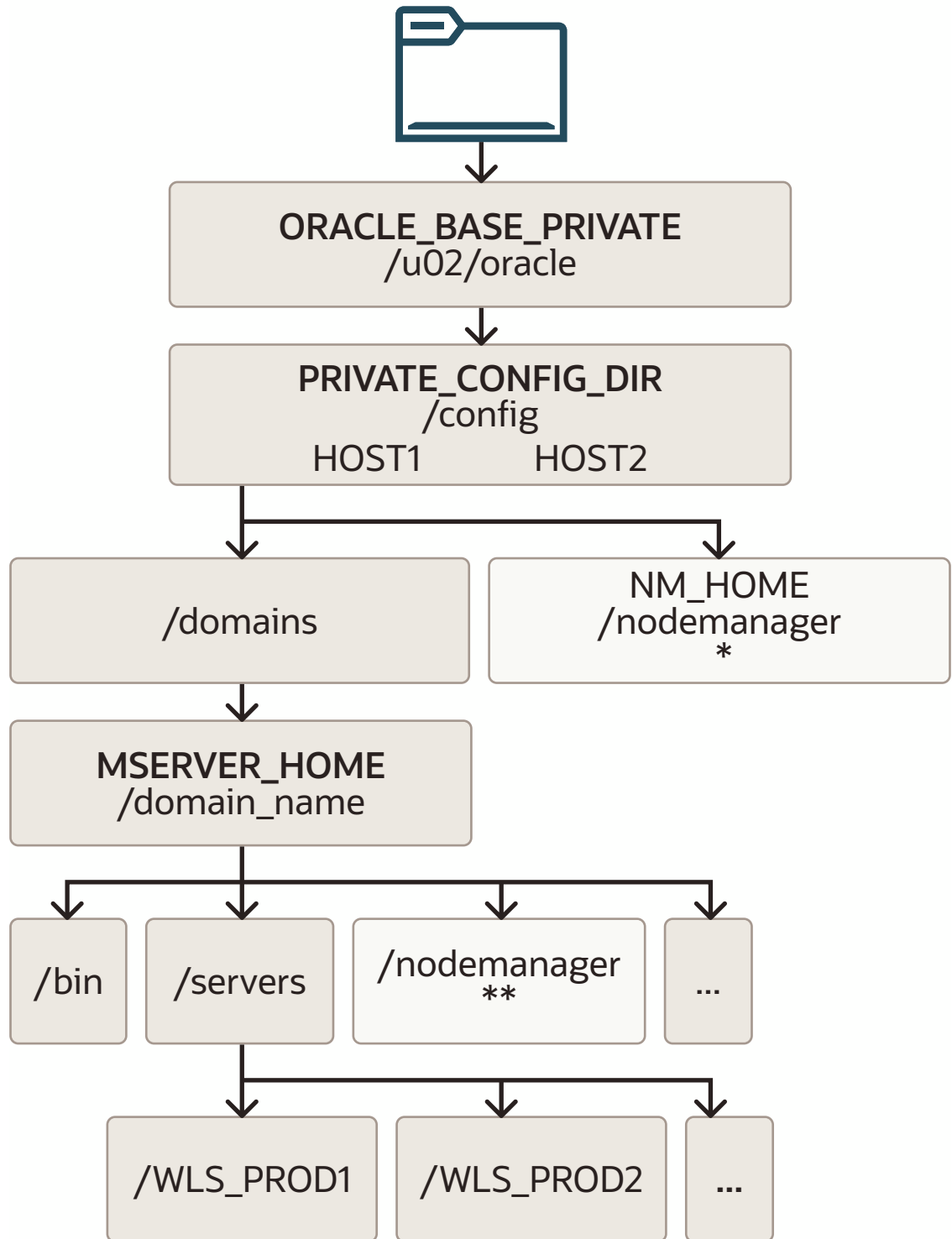
Where applicable, the diagrams also include the standard variables used to reference the directory locations in the installation and configuration procedures in this guide.

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



Legend:

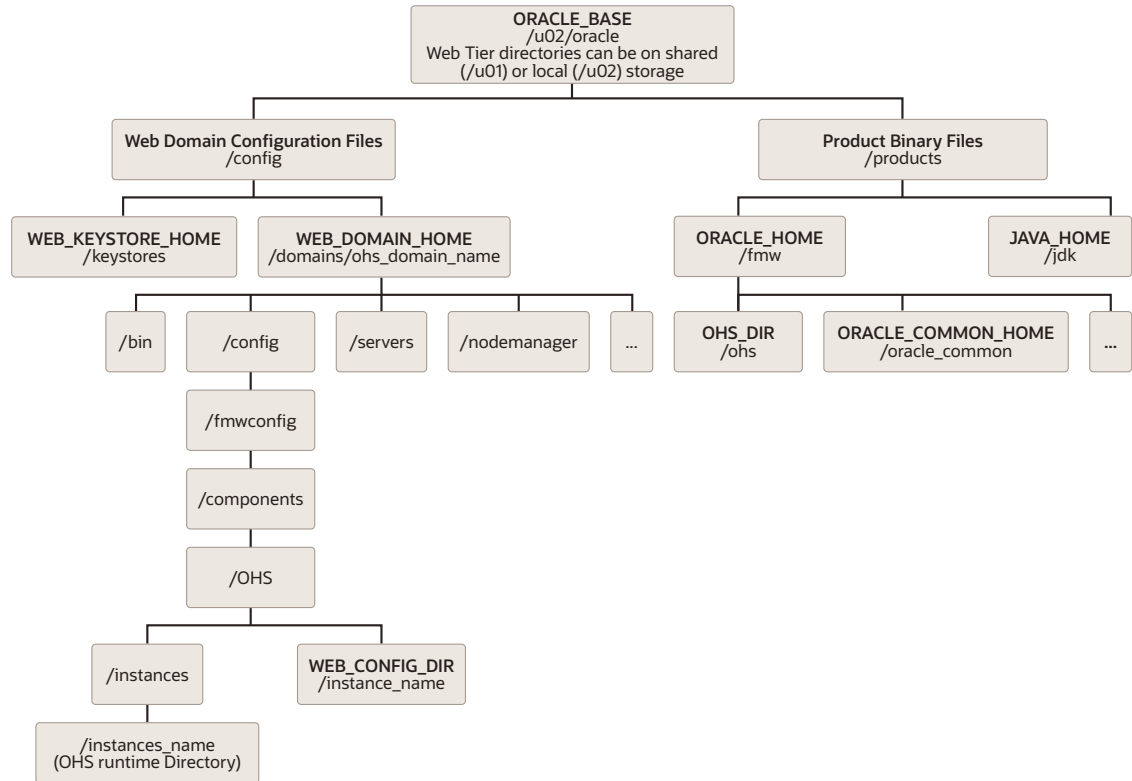
HOST1: NFS Volume 5

HOST2: NFS Volume 6

* Used for per-host Node Manager

* 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
<i>ORACLE_BASE</i>	The base directory, under which Oracle products are installed.	N/A	/u01/oracle
<i>ORACLE_HOME</i>	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.	<i>ORACLE_BASE</i> /products/fmw	/u01/oracle/products/fmw
<i>ORACLE_COMMON_HOME</i>	The directory within the Oracle Fusion Middleware Oracle home where common utilities, libraries, and other common Oracle Fusion Middleware products are stored.	<i>ORACLE_HOME</i> /oracle_common	/u01/oracle/products/fmw/oracle_common
<i>WL_HOME</i>	The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored.	<i>ORACLE_HOME</i> /wlserver	/u01/oracle/products/fmw/wlserver
<i>PROD_DIR</i>	Individual product directories for each Oracle Fusion Middleware product that you install.	<i>ORACLE_HOME</i> /prod_dir	/u01/oracle/products/fmw/prod_dir The product can be soa, wcc, idm, bi, or another value, depending on your enterprise deployment.
<i>EM_DIR</i>	The product directory used to store the Oracle Enterprise Manager Fusion Middleware Control software binaries.	<i>ORACLE_HOME</i> /em	/u01/oracle/products/fmw/em
<i>JAVA_HOME</i>	The location where you install the supported Java Development Kit (JDK).	<i>ORACLE_BASE</i> /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
<i>SHARED_CONFIG_DIR</i>	The shared parent directory for shared environment configuration files, including domain configuration, keystores, and application deployments. This will typically be the root directory for other variables in this table.	<i>ORACLE_BASE</i> /config	/u01/oracle/config
<i>PRIVATE_CONFIG_DIR</i>	The local or nfs-mounted private configuration directory unique to a given host containing the machine-specific domain directory (<i>MSERVER_HOME</i>). Directory variable: <i>PRIVATE_CONFIG_DIR</i>	/u02/oracle/config	/u02/oracle/config
<i>ASERVER_HOME</i>	The Administration Server domain home, which is installed on a shared disk.	<i>SHARED_CONFIG_DIR</i> /domains/ <i>domain_name</i>	/u01/oracle/config/domains/ <i>domain_name</i> In this example, replace <i>domain_name</i> with the name of the WebLogic Server domain.
<i>MSERVER_HOME</i>	The Managed Server domain home, which is created by using the <i>unpack</i> command on the local disk of each application tier host.	<i>PRIVATE_CONFIG_DIR</i> /domains/ <i>domain_name</i>	/u02/oracle/config/domains/ <i>domain_name</i> In this example, replace <i>domain_name</i> with the name of the WebLogic Server domain.
<i>APPLICATION_HOME</i>	The Application home directory, which is installed on shared disk, so the directory is accessible by all the application tier host computers.	<i>SHARED_CONFIG_DIR</i> /applications/ <i>domain_name</i>	/u01/oracle/config/applications/ <i>domain_name</i> In this example, replace <i>domain_name</i> with the name of the WebLogic Server domain.
<i>ORACLE_RUNTIME</i>	This directory contains application runtime artifacts, such as files generated by composites using the file and ftp adapters or any other files that are generated during the execution of an application or integration flow. Typically, you mount this directory as a separate shared file system, which is accessible by all hosts in the domain.	<i>ORACLE_BASE</i> /runtime	/u01/oracle/runtime/

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
<i>NM_HOME</i>	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 .	<i>PRIVATE_CONFIG_DIR</i> / node_manager	/u02/oracle/config/ node_manager
<i>DEPLOY_PLAN_HOME</i>	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.	<i>SHARED_CONFIG_DIR</i> /dp	/u01/oracle/config/dp
<i>KEYSTORE_HOME</i>	The shared location for custom certificates and keystores.	<i>SHARED_CONFIG_DIR</i> /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
<i>WEB_ORACLE_HOME</i>	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
<i>ORACLE_COMMON_HOME</i>	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
<i>WL_HOME</i>	The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored.	/u02/oracle/ products/fmw/wlserver
<i>PROD_DIR</i>	Individual product directories for each Oracle Fusion Middleware product that you install.	/u02/oracle/ products/fmw/ohs
<i>JAVA_HOME</i>	The location where you install the supported Java Development Kit (JDK).	/u02/oracle/ products/jdk
<i>WEB_DOMAIN_HOME</i>	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 <i>domain_name</i> with the name of the WebLogic Server domain.

Table 7-3 (Cont.) Sample Values for Key Directory Variables on the Web Tier

Directory Variable	Description	Sample Value on the Web Tier
<code>WEB_CONFIG_DIR</code>	<p>This is the location where you edit the Oracle HTTP Server configuration files (for example, <code>httpd.conf</code> and <code>moduleconf/*.conf</code>) on each web host.</p> <p>Note this directory is also referred to as the OHS Staging Directory. Changes made here are later propagated to the OHS Runtime Directory.</p> <p>See Staging and Run-time Configuration Directories in the <i>Administering Oracle HTTP Server</i>.</p>	<pre>/u02/oracle/config/ domains /domain_name/ config/fmwconfig /components/OHS/ instance/ /instance_name</pre>
<code>WEB_KEYSTORE_HOME</code>	<p>If you use Oracle HTTP Server as your web server, this is the location for custom certificates and keystores.</p>	<pre>/u02/oracle/config/ keystores</pre>

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 SOAHOST2. Be sure the directory path to the binaries on SOAHOST2 is identical to the directory path on SOAHOST1.

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 (SOAHOST1 and SOAHOST2), use the same directory path on both computers.

For example:

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

- If you configure Oracle Service Bus (OSB) in its own domain, but on the same host as Oracle SOA Suite, then you must create an additional Oracle home (`ORACLE_HOME`) for the OSB binaries, and you should mount that Oracle home separately from the SOA Oracle home.

For example, the `OSB_ORACLE_HOME` might be mounted as follows:

```
/u03/oracle/products/fmw/osb
```

- Similarly, if you configure OSB in its own domain, but on the same host as Oracle SOA Suite, then you should mount the domain directories separately from the Oracle SOA Suite domain directories.

For example, the OSB Administration Server domain directory might be mounted as follows:

```
/u03/oracle/config/domains/osb_domain_name
```

And the OSB Managed Servers domain directory might be mounted as follows:

```
/u04/oracle/config/domains/osb_domain_name
```

- If you configure Oracle Managed File Transfer(MFT) in its own domain, but on the same host as Oracle SOA Suite, then you must create an additional Oracle home (ORACLE_HOME) for the MFT binaries, and you should mount that Oracle home separately from the SOA Oracle home. You cannot install MFT in the same domain as Oracle SOA Suite.

For example, the MFT_ORACLE_HOME might be mounted as follows:

```
/u03/oracle/products/fmw/mft
```

- Similarly, if you configure MFT on the same host as Oracle SOA Suite, then you should mount the domain directories separately from the Oracle SOA Suite domain directories.

For example, the MFT Administration Server domain directory might be mounted as follows:

```
/u03/oracle/config/domains/mft_domain_name
```

And the MFT Managed Servers domain directory might be mounted as follows:

```
/u04/oracle/config/domains/mft_domain_name
```

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	SOAHOST1	/u01/oracle/products/	Storage for the product binaries to be used by SOAHOST1; this is where the Oracle home directory and product directories are installed. Used initially by SOAHOST1, but can be shared with other hosts when scaling-out the topology.
NFS Volume 2	SOAHOST2	/u01/oracle/products/	Storage for the product binaries to be used by SOAHOST2; this is where the Oracle home directory and product directories are installed. Used initially by SOAHOST2, but can be shared with other hosts when scaling-out the topology.
NFS Volume 3	SOAHOST1 SOAHOST2	/u01/oracle/config/	Administration Server domain configuration, mounted to all hosts; used initially by SOAHOST1, but can be failed over to any host.
NFS Volume 4	SOAHOST1 SOAHOST2	/u01/oracle/runtime/	The runtime directory mounted to all hosts contains application runtime artifacts such as files generated by composites using the file and ftp adapters or any other files that need to be shared by all the members of the cluster and that are generated during the execution of an application or integration flow.
NFS Volume 5	SOAHOST1	/u02/oracle/config/	Local storage for the Managed Server domain directory to be used by SOAHOST1, if the private Managed Server domain directory resides on shared storage.
NFS Volume 6	SOAHOST2	/u02/oracle/config/	Local storage for the Managed Server domain directory to be used by SOAHOST2, if the private Managed Server domain directory resides 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 7	WEBHOST1	/u02/oracle/	Local storage for the Oracle HTTP Server 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.
NFS Volume 8	WEBHOST2	/u02/oracle/	Local storage for the Oracle HTTP Server 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](#)
You should create the groups and users either locally or in your NIS or LDAP server. This user is the Oracle software owner.
- [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.
- [Configuring a Host to Use an NIS/YP Host](#)
- [Mounting the Required Shared File Systems on Each Host](#)
- [Enabling the Required Virtual IP Addresses on Each Host](#)
To prepare for the enterprise deployment, you must enable the virtual IP (VIP) address required by the WebLogic Administration Server in the node where it will run by default.

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 8-1](#) are recommended values only. Oracle recommends that you tune these values to optimize the performance of the system. See your operating system documentation for more information about tuning kernel parameters.

Kernel parameters must be set to a minimum of those in [Table 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. See the documentation for your version of the database. For example, [Configuring Kernel Parameters for Linux in *Grid Infrastructure Installation and Upgrade 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 (these values are the minimum recommended values, shown here for example only):

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

The `nfiles` 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:**

Ensure that these values are not overridden by any `.conf` file located in `/etc/security/limits.d/` folder.

4. Re-login into the host computer.
5. Use the following commands to check the current values:

```
echo "soft nfile = $(ulimit -S -n)"
echo "hard nfile = $(ulimit -H -n)"
echo "soft nproc = $(ulimit -S -u)"
echo "hard nproc = $(ulimit -H -u)"
```

Execute these commands with user 'root' and user 'oracle' to check the effective values for each user.

Verifying IP Addresses and Host Names in DNS or Hosts File

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

```
IP_Address Fully_Qualified_Name Short_Name
```

For example:

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

Oracle also recommends that you use aliases to map to different IPs in different data centers in preparation for disaster recovery. For more information about disaster recovery, see *Disaster Recovery Guide*. You can also use these aliases to configure the listen address for some of the components.

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

```
10.229.188.204 host1-vip.example.com host1-vip ADMINVHN
10.229.188.205 host1.example.com host1 SOAHOST1
10.229.188.206 host2.example.com host2 SOAHOST2
10.229.188.207 host3.example.com host3 WEBHOST1
10.229.188.208 host4.example.com host4 WEBHOST2
```

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 SOA Suite technologies, for example, are based on Unicode.

If the operating system is configured to use a non-UTF-8 encoding, Oracle SOA Suite 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

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

Configuring Users and Groups

You should create the groups and users 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 the 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 exist 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 the 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.

Since Oracle Linux 8 and Red Hat 8, the `chrony` daemon service replaces `ntpd` for the management of NTP. `Chrony` is a feature that implements NTP to maintain timekeeping accurately on the network.

To configure a host to use an NTP server with `chrony`:

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

```
server ntpost1.example.com
server ntpost2.example.com
```

4. Use the following `systemctl` command to check the status the Chrony daemon, `chronyd`:

```
systemctl status chronyd
```

5. Use the following `systemctl` command to start or restart `chronyd`:

```
systemctl restart chronyd
```

6. Run the following `chronyc -n tracking` command to check chrony tracking:

```
chronyc -n tracking
```

7. Ensure the time is set correctly using the `date` command.
8. To ensure that the server always uses the NTP server to synchronize the time, set the client to start on reboot by using the following command:

```
systemctl enable chronyd
```

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 Exalagic* in the *Oracle Fusion Middleware Exalagic Machine Owner's Guide* for more information.

After 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

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 SOAHOST1 and SOAHOST2 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 SOAHOST1, 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 SOAHOST2 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 SOAHOST1. 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,wsiz=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

To prepare for the enterprise deployment, you must enable the virtual IP (VIP) address required by the WebLogic Administration Server in the node where it will run by default.

See [Reserving the Required IP Addresses for an Enterprise Deployment](#).

This guide recommends using Service Migration instead of Server Migration for high availability of services across the members of a WLS cluster. A Virtual IP is required only for the Administration Server's listen address so that it can be failed over manually to a different node in a loss of host scenario. It is assumed that this VIP and its mapping virtual host name have been provisioned and enabled by your network administrator so that the Administration Server can use it as a valid listen address.

To enable a VIP addresses on a 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 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.
- [Implementing a Database Growth Management Strategy for Oracle SOA Suite](#)
An Oracle enterprise deployment, including Oracle SOA Suite, presents several challenges for database administrators, including managing the growth of the Oracle SOA Suite database. Underestimating the importance of managing the database can lead to issues when the database is moved to a production environment.

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](#)
- [Setting the PROCESSES Database Initialization Parameter for an Enterprise Deployment](#)

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.

Pluggable databases (PDBs) are also supported for Oracle Fusion Middleware schemas, see *Interoperability with Supported Databases in Understanding Interoperability and Compatibility*.

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 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*.

Setting the PROCESSES Database Initialization Parameter for an Enterprise Deployment

[Table 9-1](#) lists some of the typical Oracle SOA Suite enterprise topologies and the value that you should use when you set the PROCESSES initialization parameter for each topology.

Use this information as a guide when you configure the Oracle RAC database for an enterprise deployment.

Table 9-1 Required Initialization Parameters

Configuration	Parameter	Required Value (Classic Value)	Required Value (Reference Configuration Domain)	Parameter Class
SOA	PROCESSES	300 or greater	600 or greater	Static
BAM	PROCESSES	200 or greater	*	Static
SOA and BAM	PROCESSES	500 or greater	-	Static
SOA and OSB	PROCESSES	800 or greater	1200 or greater	Static

* BAM does not support Reference Configuration topology.

To check the value of the initialization parameter using SQL*Plus, you can use the `SHOW PARAMETER` command.

1. As the SYS user, issue the `SHOW PARAMETER` command as follows:

```
SQL> SHOW PARAMETER processes;
```

2. Set the initialization parameter by using the following command:

```
SQL> ALTER SYSTEM SET processes=300 SCOPE=SPFILE;
```

3. Restart the database.

The method that you use to change a parameter's value depends on whether the parameter is static or dynamic, and on whether your database uses a parameter file or a server parameter file.

 **Note:**

For more information on changing parameter values, see *Changing Initialization Parameter Values* in *Oracle Database Administrator's Guide*. For more information on database parameters for Reference Configuration Topology, see *Database Settings* under Configured Reference Configuration Domain Settings in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

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 19c 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 19c database, see *Overview of Automatic Workload Management with Dynamic Database Services* in *Real Application Clusters Administration and Deployment Guide*.

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

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

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

To create and modify a database service:

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

```
srvctl add service -db soadb -service soaedg.example.com -preferred soadb1,soadb2
```

 **Note:**

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

If the database is a multitenant database, provide the pluggable database (PDB) name when creating the service so that the service is associated with the specified PDB. For example:

```
srvctl add service -db soadb -service soaedg.example.com -preferred  
soadb1,soadb2 -pdb PDB1
```

2. Start the service:

```
srvctl start service -db soadb -service soaedg.example.com
```

 **Note:**

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

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

Use the following resources in the Oracle Database 19c *Real Application Clusters Administration and Deployment Guide* to set the SERVICE_TIME and THROUGHPUT service-level goals:

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

For example:

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

```
srvctl config service -db soadb -service soaedg.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 soadb -service soaedg.example.com -rlbgoal  
SERVICE_TIME -clbgoal SHORT
```

4. Restart the service:

```
srvctl stop service -db soadb -service soaedg.example.com  
srvctl start service -db soadb -service soaedg.example.com
```

5. Verify the change in the configuration:

```
srvctl config service -db soadb -service soaedg.example.com  
Runtime Load Balancing Goal: SERVICE_TIME  
Service name: soaedg.example.com  
Service is enabled  
Server pool: soadb_soaedg.example.com
```

```
...  
Connection Load Balancing Goal: SHORT  
Runtime Load Balancing Goal: SERVICE_TIME  
...
```

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

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

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

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:** The default setting prior to 12c. Allows SecureFile LOB storage when the `SECUREFILE` keyword is used. The default storage method is BasicFile.
- **PREFERRED:** The default setting from 12c onward, which uses SecureFile LOB storage in all cases where LOB storage would otherwise default to BasicFile.
- **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).
- **IGNORE:** Ignore attempts to create SecureFiles.
- **NEVER:** Disallow new SecureFiles creations.

The default setting for using SecureFiles from Oracle 12c Databases onward, 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 12c database or higher version.

When SOA is configured using the Reference Configuration feature (which is the configuration used for Enterprise Deployment Guide), Oracle recommends to set the `db_securefile` system parameter to "ALWAYS" on Oracle 12c databases and higher. For example:

```
sql> alter system set db_securefile=ALWAYS scope=both;
```

For more information about database settings on a Reference Configuration domain, see *Database Settings in Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

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.

Implementing a Database Growth Management Strategy for Oracle SOA Suite

An Oracle enterprise deployment, including Oracle SOA Suite, presents several challenges for database administrators, including managing the growth of the Oracle SOA Suite database. Underestimating the importance of managing the database can lead to issues when the database is moved to a production environment.

For information about determining an appropriate strategy and planning for capacity, testing, and monitoring, see [Managing Database Growth](#) in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*.

Part III

Configuring the Enterprise Deployment

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

- [Creating the Initial Infrastructure Domain for an Enterprise Deployment](#)
- [Configuring Oracle HTTP Server for an Enterprise Deployment](#)
For an enterprise deployment, Oracle HTTP Server must be installed on each of the web tier hosts and configured as Oracle HTTP standalone domains on each host.
- [Extending the Domain with Oracle SOA Suite](#)
- [Extending the Domain with Oracle Service Bus](#)
The procedures described in this chapter guide you through the process of extending the enterprise deployment topology with Oracle Service Bus (OSB).
- [Extending the Domain with Business Process Management](#)
- [Extending the Domain with Oracle Enterprise Scheduler](#)
- [Extending the Domain with Business Activity Monitoring](#)
- [Extending the Domain with Oracle B2B](#)
- [Configuring Oracle Managed File Transfer in an Enterprise Deployment](#)
The procedures explained in this chapter guide you through the process of adding Oracle Managed File Transfer to your enterprise deployment.

10

Creating the Initial Infrastructure Domain for an Enterprise Deployment

The following topics describe how to install and configure an initial domain, which can be used as the starting point for an enterprise deployment. Later chapters in this guide describe how to extend this initial domain with the various products and components that comprise the enterprise topology that you are deploying.

- [About the Initial Infrastructure Domain](#)
Before you create the initial Infrastructure domain, be sure to review the following key concepts.
- [Variables Used When Creating the Infrastructure Domain](#)
As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.
- [Installing the Oracle Fusion Middleware Infrastructure on SOAHOST1](#)
Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for an enterprise deployment.
- [Creating the Database Schemas](#)
Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.
- [Configuring the Infrastructure Domain](#)
The following topics provide instructions for creating a WebLogic Server domain using the Fusion Middleware Configuration wizard.
- [Download and Configure WebLogic Remote Console](#)
This section describes how to download and configure the WebLogic Remote Console.
- [Configuring SSL Certificates for the Domain](#)
This section describes how to configure SSL certificates for the domain.
- [Configuring a Per Host Node Manager for an Enterprise Deployment](#)
For specific enterprise deployments, Oracle recommends that you configure a per-host Node Manager, as opposed to the default per-domain Node Manager.
- [Configuring the Domain Directories and Starting the Servers on SOAHOST1](#)
After the domain is created and the node manager is configured, you can then configure the additional domain directories and start the Administration Server and the Managed Servers on SOAHOST1.
- [Configuring Web Services Manager](#)
This section describes how to configure Web Services Manager.
- [Propagating the Domain and Starting the Servers on SOAHOST2](#)
After you start and validate the Administration Server and WLS_WSM1 Managed Server on SOAHOST1, you can then perform the following tasks on SOAHOST2.
- [Modifying the Upload and Stage Directories to an Absolute Path](#)

- [Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group](#)
When you configure an Oracle Fusion Middleware domain, the domain is configured by default to use the WebLogic Server authentication provider (`DefaultAuthenticator`). However, for an enterprise deployment, Oracle recommends that you use a dedicated, centralized LDAP-compliant authentication provider.
- [Adding 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 (`SOA Administrators`) as a member to the `policy.Updater` role in the `wsm-pm` application stripe.
- [Backing Up the Configuration](#)
It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.
- [Verification of Manual Failover of the Administration Server](#)

About the Initial Infrastructure Domain

Before you create the initial Infrastructure domain, be sure to review the following key concepts.

- [About the Infrastructure Distribution](#)
- [Characteristics of the Domain](#)

About the Infrastructure Distribution

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

The Oracle JRF software consists of Oracle Web Services Manager, Oracle Application Development Framework (Oracle ADF), Oracle Enterprise Manager Fusion Middleware Control, the Repository Creation Utility (RCU), and other libraries and technologies that are required to support the Oracle Fusion Middleware products.

Later in this guide, you can then extend the domain to support the Oracle Fusion Middleware products that are required for your enterprise deployment.

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

Characteristics of the Domain

The following table lists some of the key characteristics of the domain that you are about to create. Reviewing these characteristics helps you to understand the purpose and context of the procedures that are used to configure the domain.

Many of these characteristics are described in more detail in [Understanding a Typical Enterprise Deployment](#).

Characteristic of the Domain	More Information
Uses a separate virtual IP (VIP) address for the Administration Server.	Configuration of the Administration Server and Managed Servers Domain Directories
Uses separate domain directories for the Administration Server and the Managed Servers in the domain.	Configuration of the Administration Server and Managed Servers Domain Directories
Includes a dedicated cluster for Oracle Web Services Manager	Using Oracle Web Services Manager in the Application Tier
Uses a per host Node Manager configuration.	About the Node Manager Configuration in a Typical Enterprise Deployment
Requires a separately installed LDAP-based authentication provider.	Understanding OPSS and Requests to the Authentication and Authorization Stores

Variables Used When Creating the Infrastructure Domain

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

These directory variables are defined in [File System and Directory Variables Used in This Guide](#).

- ORACLE_HOME
- ASERVER_HOME
- MSERVER_HOME
- APPLICATION_HOME
- JAVA_HOME
- NM_HOME
- KEYSTORE_HOME

In addition, you reference the following virtual IP (VIP) addresses and host names that are defined in [Physical and Virtual IP Addresses Required by the Enterprise Topology](#):

- ADMINVHN (ADMINVIP)
- SOAHOST1
- SOAHOST2
- DBHOST1
- DBHOST2
- SCAN Address for the Oracle RAC Database (DB-SCAN.example.com)

Installing the Oracle Fusion Middleware Infrastructure on SOAHOST1

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

- [Installing a Supported JDK](#)

- [Starting the Infrastructure Installer on SOAHOST1](#)
- [Navigating the Infrastructure Installation Screens](#)
- [Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers](#)
- [Checking the Directory Structure](#)
After you install the Oracle Fusion Middleware Infrastructure and create the Oracle home, you should see the directory and sub-directories listed in this topic. The contents of your installation vary based on the options that you selected during the installation.
- [Disabling the Derby Database](#)

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](#)

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:

<https://www.oracle.com/java/technologies/downloads/>

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

Installing the JDK Software

Install the JDK onto the VOL1 and VOL2 shared storage volumes mounted to `/u01/oracle/products` on the application tier hosts. Name the folder for the JDK without version numbers to avoid re-configuration challenges during JDK upgrades. Example: `/u01/oracle/products/jdk`.



Note:

Multiple installations may be needed as recommended mount points use multiple product shared volumes.

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

The following example describes how to install a recent version of JDK 17.0.

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-17.0.10+11_linux-x64_bin.tar.gz
```

 **Note:**

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 jdk-17.0.10 /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 -version
```

The Java version in the output should be displayed as `17.0.10`.

6. Repeat steps 1 through 5 for each unique *products* shared volume on an appropriate host. For example: SOAHOST1 and SOAHOST2.

Starting the Infrastructure Installer on SOAHOST1

To start the installation program, perform the following steps.

1. Log in to SOAHOST1.
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 -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_14.1.2.0.0_infrastructure.jar**.

For more information about the actual file names of each distribution, see [Identifying and Obtaining Software Downloads for an Enterprise Deployment](#).

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

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 10-1 Navigating the Infrastructure Installation Screens

Screen	Description
Installation Inventory Setup	<p>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.</p> <p>See Understanding the Oracle Central Inventory in <i>Installing Software with the Oracle Universal Installer</i>.</p>
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 <code>ORACLE_HOME</code> variable listed in Table 7-2 .
Installation Type	Use this screen to select the type of installation and as a consequence, the products and feature sets that you want to install. For this topology, select Fusion Middleware Infrastructure .

 **Note:**

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.

 **Note:**

The topology in this document does not include server examples. Oracle strongly recommends that you do not install the examples into a production environment.

Table 10-1 (Cont.) Navigating the Infrastructure Installation Screens

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

Installing Oracle Fusion Middleware Infrastructure on the Other Host Computers

If you have configured a separate shared storage volume or partition for secondary hosts, then you must install the Infrastructure on one of those hosts.

See [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

To install the software on the other host computers in the topology, log in to each host, and use the instructions in [Starting the Infrastructure Installer on SOAHOST1](#) 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](#)

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 $ORACLE_HOME
```

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

```
bin
cfgtoollogs
coherence
em
install
inventory
jlib
lib
OPatch
opmn
oracle_common
oraInst.loc
oui
wlserver
```

See [What are the Key Oracle Fusion Middleware Directories?](#) in *Understanding Oracle Fusion Middleware*.

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 Database Schemas

Oracle Fusion Middleware components require the existence of schemas in a database before you configure a Fusion Middleware Infrastructure domain. Install the schemas listed in this topic in a certified database for use with this release of Oracle Fusion Middleware.

- Metadata Services (MDS)
- Audit Services (IAU)
- Audit Services Append (IAU_APPEND)
- Audit Services Viewer (IAU_VIEWER)
- Oracle Platform Security Services (OPSS)
- User Messaging Service (UMS)
- WebLogic Services (WLS)
- Common Infrastructure Services (STB)

Use the Repository Creation Utility (RCU) to create the schemas. This utility is installed in the Oracle home for each Oracle Fusion Middleware product. For more information about RCU and how the schemas are created and stored in the database, see *Preparing for Schema Creation in Creating Schemas with the Repository Creation Utility*.

Complete the following steps to install the required schemas:

- [Installing and Configuring a Certified Database](#)
- [Starting the Repository Creation Utility \(RCU\)](#)
- [Navigating the RCU Screens to Create the Schemas](#)
- [Verifying Schema Access](#)

Installing and Configuring a Certified Database

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

See the [Preparing the Database for an Enterprise Deployment](#).

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Navigate to the `$ORACLE_HOME/oracle_common/bin` directory on your system.

```
cd $ORACLE_HOME/oracle_common/bin
```

2. Make sure that the `JAVA_HOME` environment variable is set to the location of a certified JDK on your system. The location should be up to but not including the `bin` directory. For example, if your JDK is located in `/u01/oracle/products/jdk`:

On UNIX operating systems:

```
export JAVA_HOME=/u01/oracle/products/jdk
```

3. Start RCU:

On UNIX operating systems:

```
./rcu
```

 **Note:**

If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces that are created by the RCU, provide the `-encryptTablespace true` option when you start RCU.

This defaults the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See *Encrypting Tablespaces in Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

Follow the instructions in this section to create the schemas for the Fusion Middleware Infrastructure domain:

- [Task 1, Introducing RCU](#)
- [Task 2, Selecting a Method of Schema Creation](#)
- [Task 3, Providing Database Connection Details](#)
- [Task 4, Specifying a Custom Prefix and Selecting Schemas](#)
- [Task 5, Specifying Schema Passwords](#)
- [Task 6, Verifying the Tablespaces for the Required Schemas](#)
- [Task 7, Creating Schemas](#)
- [Task 8, Reviewing Completion Summary and Completing RCU Execution](#)

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 a SQL script, which can be provided to your database administrator. See *Understanding System Load and Product Load in Creating Schemas with the Repository Creation Utility*.

Click **Next**.

 **Tip:**

For more information about the options on this screen, see *Create repository in Creating Schemas with the Repository Creation Utility*.

Task 3 Providing Database Connection Details

Provide the database connection details for RCU to connect to your database.

1. As Database Type, select **Oracle Database enabled for edition-based redefinition**.

 **Note:**

Oracle Database enabled for edition-based redefinition (EBR) is recommended to support Zero Down Time upgrades. For more information, see <https://www.oracle.com/database/technologies/high-availability/ebr.html>.

2. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
3. Enter the **Port** number of the RAC database scan listener, for example 1521.
4. Enter the RAC **Service Name** of the database.
5. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.
6. Enter the **Password** of the user name that you provided in step 4.
7. If you have selected the SYS user, ensure that you set the role to SYSDBA.
8. Click **Next** to proceed, and 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 that 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 FMW1412_

 **Tip:**

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

For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.

2. Select **AS Common Schemas**.

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

If the schemas in this section are not automatically selected, then select the required schemas.

There are two mandatory schemas that are selected by default. You cannot deselect them: **Common Infrastructure Services** (the STB schema) and **WebLogic Services** (the WLS schema). The **Common Infrastructure Services** schema enables you to retrieve information from RCU during domain configuration. See Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.

 **Tip:**

For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Click **Next** to proceed, and then click **OK** on the dialog window confirming that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords

Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database security requirements before you continue. RCU proceeds at this point even if you do not meet the password policies. Hence, perform this check outside RCU itself.

Click **Next**.

 **Tip:**

You must make a note of the passwords you set on this screen; you need them later on during the domain creation process.

Task 6 Verifying the Tablespaces for the Required Schemas

You can accept the default settings on the remaining screens, or you can customize how RCU creates and uses the required tablespaces for the Oracle Fusion Middleware schemas.

 **Note:**

You can configure a Fusion Middleware component to use JDBC stores for JMS servers and Transaction Logs, by using the Configuration Wizard. These JDBC stores are placed in the Weblogic Services component tablespace. If your environment expects to have a high level of transactions and JMS activity, you can increase the default size of the <PREFIX>_WLS tablespace to better suit the environment load.

Click **Next** to continue, and then click **OK** on the dialog window to confirm the tablespace creation.

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

Task 7 Creating Schemas

Review the summary of the schemas to be loaded and click **Create** to complete schema creation.

 **Note:**

If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and recreate the schemas before you continue.

Task 8 Reviewing Completion Summary and Completing RCU Execution

When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users are created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

 **Note:**

If the database is a pluggable database (PDB), the appropriate tns alias that points to the PDB must be used in the sqlplus command.

```
./sqlplus FMW1412_WLS/<WLS_schema_password>
```

```
SQL*Plus: Release 23.0.0.0.0 - for Oracle Cloud and Engineered Systems on Wed  
Sep 11 14:20:00 2024 Version 23.5.0.24.07  
Copyright (c) 1982, 2024, Oracle. All rights reserved.
```

```
Connected to:  
Oracle Database 23ai EE Extreme Perf Release 23.0.0.0.0 - for Oracle Cloud  
and Engineered Systems  
Version 23.5.0.24.07
```

```
SQL>
```

Configuring the Infrastructure Domain

The following topics provide instructions for creating a WebLogic Server domain using the Fusion Middleware Configuration wizard.

For more information on the other methods that are available for creating a domain, see *Additional Tools for Creating, Extending, and Managing WebLogic Domains* in *Creating WebLogic Domains Using the Configuration Wizard*.

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens to Configure the Infrastructure Domain](#)

Starting the Configuration Wizard

To begin domain configuration, run the following command in the Oracle Fusion Middleware Oracle home on SOAHOST1.

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

Navigating the Configuration Wizard Screens to Configure the Infrastructure Domain

Follow the instructions in the following sections to create and configure the domain for the topology.

- [Task 1, Selecting the Domain Type and Domain Home Location](#)
- [Task 2, Selecting the Configuration Templates](#)
- [Task 3, Selecting the Application Home Location](#)
- [Task 4, Configuring the Administrator Account](#)
- [Task 5, Specifying the Domain Mode and JDK](#)
- [Task 6, Specifying the Database Configuration Type](#)
- [Task 7, Specifying JDBC Component Schema Information](#)
- [Task 8, Providing the GridLink Oracle RAC Database Connection Details](#)
- [Task 9, Testing the JDBC Connections](#)
- [Task 10, Selecting Advanced Configuration](#)
- [Task 11, Configuring the Administration Server Listen Address](#)
- [Task 12, Configuring Node Manager](#)
- [Task 13, Configuring Managed Servers](#)
- [Task 14, Configuring a Cluster](#)
- [Task 15, Assigning Server Templates](#)
- [Task 16, Configuring Dynamic Servers](#)
- [Task 17, Assigning Managed Servers to the Cluster](#)
- [Task 18, Configuring Coherence Clusters](#)
- [Task 19, Creating Machines](#)
- [Task 20, Assigning Servers to Machines](#)
- [Task 21, Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Task 22, Writing Down Your Domain Home and Administration Server URL](#)

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:**

For more information about the other options on this screen of the Configuration Wizard, see 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 Enterprise Manager [em]**
Selecting this template automatically selects the following dependencies:
 - Oracle JRF [oracle_common]
 - WebLogic Coherence Cluster Extension [wlserver]
- **Oracle WSM Policy Manager [oracle_common]**

 **Tip:**

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

Task 3 Selecting the Application Home Location

On the Application Location screen, specify the value of the `APPLICATION_HOME` variable, as defined in [File System and Directory Variables Used in This Guide](#).

 **Tip:**

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

Task 4 Configuring the Administrator Account

On the Administrator Account screen, specify the user name (oracle recommends using a different name from “WebLogic”) and password for the default WebLogic Administrator account for the domain.

Make a note of the user name and password specified on this screen; you need these credentials later to boot and connect to the domain’s Administration Server and for other operations.

Task 5 Specifying the Domain Mode and JDK

On the Domain Mode and JDK screen:

- Select **Production** in the Domain Mode field.
In the **Enable or Disable Default Ports for You Domain** field, use the default values provided for Production Mode:
 - Ensure that **Enable Listen Ports (non-SSL Ports)** is NOT checked.

- Ensure that **Enable SSL Listen Ports** is checked.
- Ensure that **Enable Administration Port (SSL Port)** is checked.

 **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*.

- Select **Oracle Hotspot** JDK in the JDK field.

 **Note:**

Ensure that it points to the folder where you have installed the JDK. See [Installing the JDK Software](#).

Task 6 Specifying the Database Configuration Type

On the Database Configuration Type screen:

- Select **RCU Data** to activate the fields on this screen.

The **RCU Data** option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for the schemas needed to configure the domain.

- Verify that **Vendor** is `Oracle` and **Driver** is `*Oracle's Driver (Thin) for Service Connections; Versions: Any`.
- Verify that **Connection Parameters** is selected.

 **Note:**

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

After you select **RCU Data**, fill in the fields as shown in the following table:

Field	Description
Host Name	Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the <i>Enterprise Deployment Workbook</i> . For information about the Enterprise Deployment Workbook, see Using the Enterprise Deployment Workbook .

Field	Description
DBMS/Service	<p>Enter the service name for the Oracle RAC database appropriate for this domain where you will install the product schemas. For example:</p> <p><code>soaedg.example.com</code></p> <p>Specify the service name based on the value configured earlier in the Preparing the Database for an Enterprise Deployment section.</p>
Port	Enter the port number on which the database listens. For example, 1521.
Schema Owner Schema Password	<p>Enter the user name and password to connect to the database's Service Table schema.</p> <p>This is the schema user name and password that was specified for the Service Table component on the <i>Schema Passwords</i> screen in RCU (see Creating the Database Schemas).</p> <p>The default user name is <code>prefix_STB</code>, where <code>prefix</code> is the custom prefix that you defined in RCU.</p>

Click **Get RCU Configuration** when you finish specifying the database connection information. The following output in the Connection Result Log indicates that the operation is successful.

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Click **Next** if the connection to the database is successful.

 **Tip:**

More information about the **RCU Data** option can be found in Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*. More information about the other options on this screen can be found in Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 7 Specifying JDBC Component Schema Information

Verify that the values on the JDBC Component Schema screen are correct for all schemas. The schema table should be populated, because you selected **Get RCU Data** on the previous screen. As a result, the Configuration Wizard locates the database connection values for all the schemas required for this domain.

At this point, the values are configured to connect to a single-instance database. However, for an enterprise deployment, you should use a highly available Real Application Clusters (RAC) database, as described in [Preparing the Database for an Enterprise Deployment](#).

In addition, Oracle recommends that you use an Active GridLink datasource for each of the component schemas. For more information about the advantages of using GridLink data sources to connect to a RAC database, see Database Considerations in the *High Availability Guide*.

To convert the data sources to GridLink:

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

Task 8 Providing the GridLink Oracle RAC Database Connection Details

On the GridLink Oracle RAC Component Schema screen, provide the information required to connect to the RAC database and component schemas, as shown in following table.

Element	Description and Recommended Value
Service Name	Verify that the service name for the Oracle RAC database is the appropriate. For example, <i>soaedg.example.com</i> .
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	These values are not required when you are using an Oracle 12c database or higher versions because the ONS list is automatically provided from the database to the driver.
Enable Fan	Verify that the Enable Fan check box is selected, so the database can receive and process FAN events.

For more information about specifying the information on this screen, as well as information about how to identify the correct SCAN address, see *Configuring Active GridLink Data Sources with Oracle RAC* in the *High Availability Guide*.

You can also click **Help** to display a brief description of each field on the screen.

Task 9 Testing the JDBC Connections

Use the JDBC Component Schema Test screen to test the data source connections that 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, and then try to test the connection again.

Tip:

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

Task 10 Selecting Advanced Configuration

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

- **Administration Server**
This is required to configure the listen address of the Administration Server.
- **Node Manager**
This is required to configure Node Manager.
- **Topology**

This is required to add, delete, or modify the Settings for Server Templates, Managed Servers, Clusters, Virtual Targets, and Coherence.

 **Note:**

When you use 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 that you selected the required templates for this topology.
- Do not select the **Domain Frontend Host Capture** advanced configuration option. You later configure the frontend host property for specific clusters, rather than for the domain.

Task 11 Configuring the Administration Server Listen Address

On the Administration Server screen:

1. In the **Server Name** field, retain the default value "AdminServer".
2. In the **Listen Address** field, enter the virtual host name that corresponds to the VIP of the ADMINVHN that you procured in Procuring Resources for an Enterprise Deployment and enabled in Preparing the Host Computers for an Enterprise Deployment.

For more information about the purpose of using the ADMINVHN virtual host, see [Reserving the Required IP Addresses for an Enterprise Deployment](#).

3. In the **Configure Administration Server Ports** section, perform the following steps:
 - a. Leave the **Enable Listen Port** field unchecked. The **Listen Port** value will be disabled in grey.
 - b. Ensure the **Enable SSL Listen port** field is checked.
 - c. Leave the default value as 7002 in the **SSL Listen Port** field.
 - d. Leave the default value as 9002 in the **Administration Port**.
4. Leave the default value as Unspecified in the **Server Group**.

Task 12 Configuring Node Manager

Select **Manual Node Manager Setup** as the Node Manager type.

 **WARNING:**

You can ignore the warning in the bottom pane. This guide provides the required steps for the Manual Node Manager configuration.

 **Tip:**

For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.

For more information about per domain and per host Node Manager implementations, see [About the Node Manager Configuration in a Typical Enterprise Deployment](#).

For information about Node Manager configurations, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Task 13 Configuring Managed Servers

Use the Managed Servers screen to create two new Managed Servers:

1. Click the **Add** button to create a new Managed Server.
2. Specify `WLS_WSM1` in the **Server name** column.
3. In the **SSL Listen Address** column, enter `SOAHOST1`. Ensure you enter the host name that corresponds to `SOAHOST1` and do not use the IP address.
4. Ensure that **Enable Listen** is unchecked and **Listen Port** is "Disabled" (grayed out).
5. Ensure that **Enable SSL Port** is checked for all servers.
6. Set **SSL Listen Port** to 7010.
7. Set **Administration Port** to 9003.
8. In the **Server Groups** drop-down list, select **JRF-MAN-SVR** and **WSMPM-MAN-SVR**.

These server groups ensure that the Oracle JRF and Oracle Web Services Manager (OWSM) services are targeted to the Managed Servers that you are creating.

Server groups target Fusion Middleware applications and services to one or more servers by mapping defined groups of application services to each defined server group. Any application services that are mapped to a given server group are automatically targeted to all servers that are assigned to that group. See Application Service Groups, Server Groups, and Application Service Mappings in *Domain Template Reference*.

 **Note:**

Nonce caching for Oracle Web Services is initialized automatically by the WSM-CACHE-SVR server group and is suitable for most custom applications. This initialization is automatically performed in SOA, OSB, and other FMW servers that run JRF and create a coherence cluster. Nonce is a unique number that can be used only once in a SOAP request and is used to prevent replay attacks. Nonce caching naturally scales with the number of added Managed Servers that run Web service applications.

For information about advanced caching configurations, see Caching the Nonce with Oracle Coherence in *Securing Web Services and Managing Policies with Oracle Web Services Manager*, which provides additional guidance for the use of nonce caching and the WSM-CACHE-SVR server-group in custom WLS servers.

9. Repeat this process to create a second Managed Server named `WLS_WSM2`.

Server Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Administration Port	Server groups
WLS_WS M1	SOAHOS T1	unchecked	Disabled	Checked	7010	9003	JRF-MAN-SVR and WSMPM-MAN-SVR
WLS_WS M2	SOAHOS T2	unchecked	Disabled	Checked	7010	9003	JRF-MAN-SVR and WSMPM-MAN-SVR

The Managed Server names suggested in this procedure (WLS_WSM1 and WLS_WSM2) are referenced throughout this document; if you choose different names then be sure to replace them as needed.

 **Tip:**

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

Task 14 Configuring a Cluster

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `WSM-PM_Cluster` in the **Cluster Name** field.
3. Click **Next**.

 **Note:**

If you specify a front-end host and a front-end port, the URL validation fails after domain configuration because the web tier is not setup at this point. Hence, any redirections in the hosted application returns to the front-end address. You must configure the web tier to allow accessing the URLs through LBR. You can configure the front-end port and address at a later point. For instructions, see [Setting the Front End Host and Port for a WebLogic Cluster](#).

 **Tips:**

For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 15 Assigning Server Templates

Click **Next**.

Task 16 Configuring Dynamic Servers

Verify that all dynamic server options are disabled for clusters that are to remain as static clusters.

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

Task 17 Assigning Managed Servers to the Cluster

Use the Assign Servers to Clusters screen to assign `WLS_WSM1` and `WLS_WSM2` to the new cluster `WSM-PM_Cluster`:

1. In the **Clusters** pane, select the cluster to which you want to assign the servers; in this case, `WSM-PM_Cluster`.
2. In the **Servers** pane, assign `WLS_WSM1` to `WSM-PM_Cluster` by doing one of the following:
 - Click once on `WLS_WSM1` to select it, and then click on the right arrow to move it beneath the selected cluster (`WSM-PM_Cluster`) in the Clusters pane.
 - or
 - Double-click on `WLS_WSM1` to move it beneath the selected cluster (`WSM-PM_Cluster`) in the clusters pane.
3. Repeat these steps to assign the `WLS_WSM2` Managed Server to the `WSM-PM_Cluster`.

Tip:

More information about the options on this screen can be found in Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain.

In the **Cluster Listen Port**, enter 9991.

Note:

For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 19 Creating Machines

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

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

Use the values in [Table 10-3](#) to define the Name and Node Manager Listen Address of each machine.

3. Verify the port in the Node Manager Listen Port field.

The port number 5556, shown in this example, may be referenced by other examples in the documentation. Replace this port number with your own port number, as needed.

Name	Node Manager Listen Address	Node Manager Type	Node Manager Listen Port
ADMINHOST	Enter the value of the ADMINVHN variable.	SSL	5556
SOAHOST1	The value of the SOAHOST1 host name variable or SOAHOST1 alias. For example, SOAHOST1.example.com.	SSL	5556
SOAHOST2	The value of the SOAHOST2 host name variable or SOAHOST2 alias. For example, SOAHOST2.example.com.	SSL	5556

 **Tip:**

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

Task 20 Assigning Servers to Machines

Use the Assign Servers to Machines screen to assign any statically defined managed servers to the appropriate machines. Servers that are part of a dynamic cluster are assigned to the calculated machine names automatically.

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 server name 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_WSM1 Managed Server to the SOAHOST1 machine.
- Assign the WLS_WSM2 Managed Server to the SOAHOST2 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 21 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the 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 22 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen shows the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you need them later; the domain location is needed to access the scripts that are used to start the Administration Server.

Click **Finish** to dismiss the Configuration Wizard.

Download and Configure WebLogic Remote Console

This section describes how to download and configure the WebLogic Remote Console.

Note:

For the initial configuration steps required in this EDG, you will need access to the AdminServer listen address and administration port. Later on you will configure access from a frontend load balancer.

Perform the following steps to download and configure the WebLogic Remote Console:

1. Uninstall any previous versions of the WebLogic Remote Console from your computer.
2. Download the WebLogic Remote Console. Go to <https://github.com/oracle/weblogic-remote-console/releases> and download the installer from your operating system.
3. Run the installer.
4. Install the WebLogic Remote Console extension in the WebLogic Server domain. The WebLogic Remote Console extension provides additional functionality when using the WebLogic Remote Console to manage WebLogic domains.

Note:

This step is optional.

- a. Create a `management-services-ext` directory under the domain home.
 - b. Download the latest WebLogic Remote Console extension, `console-rest-ext-
<version>.war`, from <https://github.com/oracle/weblogic-remote-console/releases> and save it inside the `management-services-ext` directory you created in the previous step. If you have an earlier version of the extension already downloaded, delete it and replace it with the latest version.
 - c. Reboot the Administration Server if it is already running.
5. Launch the WebLogic Remote Console application.

Example:

```
./weblogic-remote-console
```

In the next steps you must connect to the EDG domain provider using initially the Admin Servers listen address.

Configuring SSL Certificates for the Domain

This section describes how to configure SSL certificates for the domain.

- [Creating Certificates and Certificate Stores for the WebLogic Domain](#)
- [Adding Certificate Stores Location to the WebLogic Servers Start Scripts](#)
- [Update Server's Security Settings Using the Remote Console](#)
- [Configuring KSS with Per-domain CA](#)

Creating Certificates and Certificate Stores for the WebLogic Domain

The Enterprise Deployment Guide provides steps to configure a domain that uses SSL listen addresses for all Weblogic Managed Servers, Weblogic Administration Server and Node Managers in the application tier. To achieve this the required certificates for all servers, machines and NM listen addresses must be created and pointed to from the domain and Node Manager configuration.

In Oracle FMW 14.1.2.0, Oracle WebLogic allows the usage of a per-domain Certificate Authority (CA). In this model, the `CertGen` and `ImportPrivateKey` utilities are enhanced to use the domain's secret key to encrypt the passphrases and store them in the domain's `DemoCerts.props` file. A self-signed Demo CA is automatically created for the domain and it is used for signing certificates for the SSL listen addresses used in the EDG. Although in a real production system, standard CAs should be used, the per-domain CA model implements an SSL system using domain specific CA that provides a higher degree of protection than non-ssl configurations. If you want to use your own custom certificates, see [About Using Third Party SSL Certificates in the WebLogic and Oracle HTTP Servers](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Oracle recommends using a shared storage location (protected with the appropriate snapshot or file backup tooling) where all the different certificates and stores can be found by the different servers. Perform the following steps to generate an Identity store and a Trust Store that can be used for enabling SSL listeners in a Weblogic Server using a per-domain CA:

1. Download the `generate_perdomainCACERTS.sh` script in the `maa` github repo.

```
https://github.com/oracle-samples/maa/blob/main/1412EDG/  
generate\_perdomainCACERTS.sh
```

2. Run the script with the following arguments:

- `WLS_DOMAIN_DIRECTORY`: Directory hosting the WebLogic Domain that the Administration Server uses.
- `WL_HOME`: The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored. Typically `/u01/oracle/products/fmw/wlserver`.
- `KEYSTORE_HOME`: Directory where `appIdentity` and `appTrust` stores will be created.
- `KEYPASS`: Password used for the weblogic administration user (will be reused for certs and stores).

Example:

```
./generate_perdomainCACERTS.sh $ASERVER_HOME $ORACLE_HOME/  
wlserver $KEYSTORE_HOME <keypass>
```

The script will traverse the `WLS_DOMAIN_DIRECTORY/config/config.xml` to find all the listen addresses used in the domain, generate certificates for all of them, create a trust store with the domain CA and import certificates into a new Identity store. The aliases used in the import will be the same as the hostname used as listen address. Both the trust store and the identity store will be placed in the `KEYSTORE_HOME` directory.

Run the following command to verify if the "domainca" entry is there as a `trustedCertEntry`:

```
keytool -list -keystore $KEYSTORE_HOME/appTrustKeyStore.pkcs12
```

Run the following command to verify if there is a `PrivateKeyEntry` for each listen address (the values for `ADMINVHN`, `SOAHOST1` and `SOAHOST2`):

```
keytool -list -keystore $KEYSTORE_HOME/appIdentityKeyStore.pkcs12
```

Adding Certificate Stores Location to the WebLogic Servers Start Scripts

Once the Identity and Trust Stores are created for the domain some Java properties must be added to the WebLogic start scripts. These properties are added to the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Any customizations you add to this file are preserved during domain upgrade operations and are carried over to remote servers when using the `pack` and `unpack` commands.

- If you created the Identity and Trust Stores with the script `generate_perdomainCACERTS.sh`, as explained in [Creating Certificates and Certificate Stores for the WebLogic Domain](#), then the properties are automatically added to the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Just verify that the file exists and that the `EXTRA_JAVA_PROPERTIES` have been added.
- If you are using your own custom certificates, then manually create the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Edit the file and add the variable `EXTRA_JAVA_PROPERTIES` to set the `javax.net.ssl.trustStore` and `javax.net.ssl.trustStorePassword` properties with the values used by your EDG system. For example:

```
EXTRA_JAVA_PROPERTIES="{EXTRA_JAVA_PROPERTIES}  
-Djavax.net.ssl.trustStore=/u01/oracle/config/keystores/  
appTrustKeyStore.pkcs12
```

```
-Djavax.net.ssl.trustStorePassword=mypassword"  
export EXTRA_JAVA_PROPERTIES
```

 **Note:**

The order of the extra java properties is relevant. In case that the same property is defined more than once, the later value is used. The custom values must be defined as in the example provided.

Update Server's Security Settings Using the Remote Console

- [Connecting to the Remote Console Using the Administration Server's Virtual Hostname as Provider](#)
- [Updating the WebLogic Servers Security Settings](#)

Connecting to the Remote Console Using the Administration Server's Virtual Hostname as Provider

The following procedure temporarily starts the Administration Server with the default start script so to enable you to perform these tasks. After you perform these tasks, you can stop this temporary session and use the Node Manager to start the Administration Server.

 **Note:**

For this Remote Console initial access to the Administration Server, it is required that the machine that runs the Remote Console can resolve and connect to the Admin Server's Listen Address. This can be done by starting the Remote Console directly in the node where the Admin Server runs or creating a tunnel to this address from the node where the remote Console is executed.

1. Using the following default start script to start the Administration Server:
 - a. Change directory to the following directory:

```
cd $ASERVER_HOME/bin
```

- b. Run the start script:

```
./startWebLogic.sh
```

Monitor the terminal till the following message is displayed:

```
<Server state changed to RUNNING>
```

Also you must verify that the appropriate SSL listener is available, which can be confirmed with the a message like the following displayed in output:

```
<Server> <BEA-002613> <Channel "DefaultSecure" is now listening on  
XXXX:7002 for protocols iiops, t3s, ldaps, https.>
```

2. Create a new provider in the WebLogic Remote Console as follows:
 - a. Download the domain's trust keystore to the host or laptop where you run the WebLogic Remote Console. For example, when using the per-domain CA steps in previous sections, this would be located at `$KEYSTORE_HOME/appTrustKeyStore.pkcs12`.
 - b. Open the Remote Console and add the domain trust store to the remote console settings. Click **File > Settings** and enter the following values.
 - i. Trust Store type - jks
 - ii. Trust Store Path - The path to the trust keystore file in the host where the Remote Console runs.
 - iii. Trust Store Key - Enter the password provided in the steps above for certificate creation.
 - iv. Check **Disable HostName verification** if you are using Demo certificates as described in the steps above.
 - c. Using the Providers window in the Remote Console, create a new provider by selecting **Add Admin Server Connection Provider**.
 - i. In the provider name, enter the name of `soaedg_domain_asvip`. This will identify the type of access.
 - ii. Enter the WebLogic Domain Administration username provided in the configuration wizard user name.
 - iii. Enter the password used for the domain creation.
 - iv. Use https protocol and the admin server listen address used in the configuration wizard as URL for access and specify port 9002.
For example, `https://ADMINVHN.example.com:9002`.
 - v. Check the **Make Insecure Connection** checkbox.



Note:

This provider should not be used once the front end and webtier are configured.

The Remote Console Home Window for the domain will be displayed.

Updating the WebLogic Servers Security Settings

Perform the following steps to update the WebLogic Servers Security Settings and Administration Port:

1. Access the Domain provider in the Remote Console and update the Administration Server and WebLogic Servers Security Settings:
 - a. Click **Edit Tree**.

- b. Click **Environment > Servers > AdminServer**.
- c. Click **Security** tab.
- d. Change the keystores dropdown to **Custom Identity and Custom Trust**.
- e. In **Custom Identity Keystore**, enter the fully qualified path to the identity keystore as follows:

`KEYSTORE_HOME/appIdentityKeyStore.pkcs12`

Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the [Table 7-2](#).

- f. Set the **Custom Identity Keystore Type** to JKS.

 **Note:**

Specifying JKS or PKCS12 is valid both for pkcs12 and jks stores. Both formats can be read and managed if the Customer Key Store Type is set to "JKS".

- g. In **Custom Identity Keystore Passphrase**, enter the password `Keystore_Password` you provided in the certificate generation steps.

- h. In **Custom Trust Keystore**, enter the fully qualified path to the trust keystore.

`KEYSTORE_HOME/appTrustKeyStore.pkcs12`

Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the [Table 7-2](#).

- i. Set the **Custom Trust Keystore Type** to JKS.

 **Note:**

Specifying JKS or PKCS12 is valid both for pkcs12 and jks stores. Both formats can be read and managed if the Customer Key Store Type is set to "JKS".

- j. In **Custom Trust Keystore Passphrase**, enter the password you provided as the **<keypass>** in the certificate generation steps.

- k. Click **Save**.

- l. Under **Security** settings, navigate to **SSL** tab.

- m. In the **Server Private Key Alias** field enter the alias provided in the certificate generation steps. If you used the certificate generation script this will be the same as the listen address used for the WLS server.

- n. In the **Server Private Key Pass Phrase** field, enter the password provided in the certificate generation steps. If you used the certificate generation script this will be the same as the keystore passphrase.

- o. Click **Save**.

The cart on the top right part of the screen will show **full** with a yellow bag inside.

- p. Click the Cart icon on the top right and select **Commit Changes**.

Repeat the above steps for each managed server in the domain changing the alias to match the alias used for the certificates.

2. Return to the terminal window where you started the Administration Server with the start script.
3. Press **Ctrl+C** to stop the Administration Server process.
Wait for the Administration Server process to end and for the terminal command prompt to appear.

4. Start the Administration Server again by using the following script:

- a. Change directory to the following directory:

```
cd $ASERVER_HOME/bin
```

- b. Run the start script:

```
./startWebLogic.sh
```

- c. Monitor the output in the terminal till the following output is displayed.

```
<Server state changed to RUNNING>
```

Configuring KSS with Per-domain CA

For consistency purposes and to use a common CA all across the domain artifacts you may want to use the per-domain CA for KSS (store used by OPSS and other components in the WebLogic Infrastructure/JRF Domain).

Perform the following steps to import the domain CA certificate in the KSS trusted store:

1. Download the `import-domainca-into-kss.sh` script in the maa github repo <https://github.com/oracle-samples/maa/blob/main/1412EDG/import-domainca-into-kss.sh>.

2. Edit the script and customize the following variables according to your environment:

DOMAIN_HOME: Path to the WebLogic domain (`ASERVER_HOME` in this guide). For example, `/u01/oracle/config/domains/soaedg_domain`.

MW_HOME: The path to the FMW home. For example, `/u01/oracle/products/fmw`.

ADMINVHN: Administration Server's listen address. For example, `adminvhn.example.com`.

ADMINPORT: Administration Server's listen port. For example, `9002`.

DOMAINUSER: Name of the administrator user for the WLS domain. For example, `soaedgadmin`.

TRUSTSTOREFILE: Location of the truststore used to connect through SSL to the Admin Server. For example, `/u01/oracle/config/keystores/appTrustKeyStore.pkcs12`.

3. Run the script with the following arguments:

- **DOMAINPASS:** WLS domain administrator user's password
- **KEYPASS:** Password for the truststore.

Example

```
./import-domainca-into-kss.sh adminpassword123 truststorepassword123
```

The script imports the per Domain CA certificate into KSS and assigns it to jps.

You can verify that the update was successful by inspecting the jps configuration files.

```
grep domainca $ASERVER_HOME/config/fmwconfig/jps-config.xml
```

The result of the command must be similar to the following example:

```
<property name="ca.key.alias" value="domainca-new-24-05-07-16-44-52"/>
```

4. Restart the Admin Server.

If Admin Server was started with the script, perform the following steps:

- a. Press **Ctrl+C** to stop the Administration Server process.
- b. Go to directory `$ASERVER_HOME/bin` and run the following command:

```
./startWebLogic.sh
```

Configuring a Per Host Node Manager for an Enterprise Deployment

For specific enterprise deployments, Oracle recommends that you configure a per-host Node Manager, as opposed to the default per-domain Node Manager.

For more information about the advantages of a per host Node Manager, see [About the Node Manager Configuration in a Typical Enterprise Deployment](#)

- [Creating a Per Host Node Manager Configuration](#)
- [Starting the Node Manager on SOAHOST1](#)
- [Configuring the Node Manager Credentials](#)
- [Enrolling the Domain with NM](#)
- [Adding Truststore Configuration to Node Manager](#)

Creating a Per Host Node Manager Configuration

The step in configuring a per-host Node Manager is to create a configuration directory and two new node manager configuration files. You must also edit the default `startNodeManager.sh` file.

To create a per-host Node Manager configuration, perform the following tasks, first on SOAHOST1, and then on SOAHOST2:

1. Log in to SOAHOST1 and create a directory for the Node Manager configuration files :

For example:

```
mkdir -p /u02/oracle/config/nodemanager
```

Note that this directory should be on a local disk, because it is specific to the host. This directory location is known as the Node Manager home, and it is identified by the `NM_HOME` directory variable in examples in this guide.

2. Change directory to the Node Manager home directory:

```
cd $NM_HOME
```

3. Create a new text file called `nodemanager.properties` and add the values shown in [Example: Contents of the nodemanager.properties File](#) to this new file.

Use the pertaining identity alias for the node that you are configuring. For example, `soahost1.example.com` in SOAHOST1 and `soahost2.example.com` in SOAHOST2.

For more information about the properties that you can add to the `nodemanager.properties` file, see *Node Manager Properties in Administering Node Manager for Oracle WebLogic Server*.

In the `nodemanager.properties` file, you enable crash recovery for servers as a part of this configuration. See *Node Manager and System Crash Recovery in Administering Node Manager for Oracle WebLogic Server*.

Example: Contents of the `nodemanager.properties` File

```
DomainsFile=/u02/oracle/config/nodemanager/nodemanager.domains
LogLevel=INFO
LogLimit=0
PropertiesVersion=14.1.2.0.0
AuthenticationEnabled=true
NodeManagerHome=/u02/oracle/config/nodemanager
#Include the specific JDK home
JavaHome=/u01/oracle/products/jdk
LogLevel=INFO
DomainsFileEnabled=true
StartScriptName=startWebLogic.sh
#Leave blank for listening on ANY
ListenAddress=
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
SecureListener=true
LogCount=1
StopScriptEnabled=false
QuitEnabled=false
LogAppend=true
StateCheckInterval=500
CrashRecoveryEnabled=true
StartScriptEnabled=true
LogFile=/u02/oracle/config/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityAlias=soahost1.example.com
CustomIdentityKeyStoreFileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12
CustomIdentityKeyStorePassPhrase=password
CustomIdentityPrivateKeyPassPhrase=password
```

Notice the values for `CustomIdentityAlias`. If you used the `generate_perdomainCACERTS.sh` script, this is the hostname used as listen address in the configuration wizard for the Node Manager Machine. If you created the certificates one by one, this would be the alias that you assigned to the certificate import for SOAHOST1. You must also provide the location of the `IdentityStore` generated in the previous steps and the password for the same.

4. Locate the `startNodeManager.sh` file in the following directory:

```
$WL_HOME/server/bin
```

5. Copy the `startNodeManager.sh` file to the Node Manager home directory.

```
cp $WL_HOME/server/bin/startNodeManager.sh $NM_HOME
```

6. Edit the new `startNodeManager.sh` file and add the `NODEMGR_HOME` property as follows:

```
NODEMGR_HOME="NM_HOME"
```

In this example, replace *NM_HOME* with the actual path to the Node Manager home.

7. Locate the `stopNodeManager.sh` script in the `WL_HOME/server/bin` directory. Copy it to the Node Manager home directory. Edit the copied file and edit the `NODEMGR_HOME` property pointing to the node manager home (as it has been done for the `startNodemanager.sh` file):

```
NODEMGR_HOME="NM_HOME"
```

In this example, replace *NM_HOME* with the actual path to the Node Manager home.

8. Create another new file in the Node Manager home directory, called `nodemanager.domains`.

The `nodemanager.domains` file provides additional security by restricting Node Manager client access to the domains listed in this file.

9. Perform steps 1 through 8 on SOAHOST2.
10. Add the following entries to the new `nodemanager.domains` files:

On SOAHOST1, add values for both the Administration Server domain home and the Managed Servers domain home:

```
soaedg_domain=MSERVER_HOME;ASERVER_HOME
```

Note:

The path that is mentioned first (*MSERVER_HOME*) is considered as the `primaryDomainPath` and Managed Servers are run from this location.

On SOAHOST2, add the value for the Managed Servers domain home only:

```
soaedg_domain=MSERVER_HOME
```

In these examples, replace *ASERVER_HOME* and *MSERVER_HOME* with the values of the respective variables, as described in [File System and Directory Variables Used in This Guide](#).

Starting the Node Manager on SOAHOST1

After you manually set up the Node Manager to use a per-host Node Manager configuration, you can start the Node Manager on SOAHOST1, by using the `startNodeManager.sh` script.

To start the Node Manager on SOAHOST1:

1. Change directory to the Node Manager home directory:

```
cd $NM_HOME
```

2. Run the following command to start the Node Manager and send the output of the command to an output file, rather than to the current terminal shell:

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

3. Monitor the the `nodemanager.out` file; make sure the NodeManager starts successfully. The output should eventually contain the following strings:

```
<INFO> <Upgrade> <Encrypting NodeManager property:
CustomIdentityKeyStorePassPhrase>
<INFO> <Upgrade> <Encrypting NodeManager property:
CustomIdentityPrivateKeyPassPhrase>
<Upgrade> <Saving upgraded NodeManager properties to '/u02/oracle/config/
nodemanager/nodemanager.properties'>
<INFO> <Loading domains file: /u02/oracle/config/nodemanager/
nodemanager.domains>
<INFO> <Loading identity key store: FileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12, Type=pkcs12, PassPhraseUsed=true>
<INFO> <Loaded NodeManager configuration properties from '/u02/oracle/
config/nodemanager/nodemanager.properties'>
<INFO> <14.1.2.0.0>
<INFO> <Server Implementation Class:
weblogic.nodemanager.server.NMServer$ClassicServer.>
<INFO> <Secure socket listener started on port 5556>
```

You must check that the plain text used for passwords in `nodemanager.properties` has now been encrypted:

```
[oracle@soalonhost1 keystores]$ cat /u02/oracle/config/nodemanager/
nodemanager.properties
#Tue Feb 06 11:53:10 GMT 2024
#Mon Feb 05 17:24:30 GMT 2024
DomainsFile=/u02/oracle/config/nodemanager/nodemanager.domains
LogLimit=0
PropertiesVersion=14.1.2.0.0
AuthenticationEnabled=true
NodeManagerHome=/u02/oracle/config/nodemanager
#Include the specific JDK home
JavaHome=/u01/oracle/products/jdk
LogLevel=INFO
DomainsFileEnabled=true
StartScriptName=startWebLogic.sh
#Leave blank for listening on ANY
ListenAddress=
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
SecureListener=true
LogCount=1
StopScriptEnabled=false
QuitEnabled=false
LogAppend=true
StateCheckInterval=500
CrashRecoveryEnabled=true
StartScriptEnabled=true
LogFile=/u02/oracle/config/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
```

```
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityAlias=soahost1.example.com
CustomIdentityKeyStoreFileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12
CustomIdentityKeyStorePassPhrase={AES256}EMvPrOCRfN7fyv3d8JcEnttTLyneG9Su+U
VK5DGEmqmqDwLkpLz9nQFZ+fL1Bidc
CustomIdentityPrivateKeyPassPhrase={AES256}O5cEJD8WVYP3aRLp9KAbFZ3CGLyxmmIW
FX1YzVfJvPp11dc5RbMksAcsBLquKcWW
```

Configuring the Node Manager Credentials

Perform the following steps to set the Node Manager credentials using the Remote Console:

1. Access the Domain provider in the Remote Console.
2. Click **Edit Tree**.
3. Click **Environment > Domain > Security**.
4. Check the **Show Advanced Fields** field.
5. Set **Node Manager Username** to the same as the Weblogic Administrator, since this username will be used in other tasks mentioned in this guide.
6. Change the NM password. Ensure the **Node Manager password** is set to the same as the Weblogic Administrator since this password will be used in other tasks mentioned in this guide.
7. Click **Save**. The cart on the top right part of the screen will show **full** with a yellow bag inside.
8. Click the Cart Icon on the top right and select **Commit Changes**.

Enrolling the Domain with NM

Perform the following steps in a new terminal window to enroll the domain with Node manager.

 **Note:**

You will be unable to connect to the Node Manager and use it to start the servers in the domain without performing this step.

1. Change directory to the following directory:

```
cd $ORACLE_COMMON_HOME/common/bin
```

2. Start the WebLogic Server Scripting Tool (WLST). In order to use the certificates created for the appropriate SSL handshake, the location of the stores and password of the same need to be provided to WLST. Use the following command or add these in a script that can be easily invoked:

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=CustomTrust -
Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/
keystores/appTrustKeyStore.pkcs12 -
```

```
Dweblogic.security.CustomTrustKeyStorePassPhrase=storepassword"
./wlst.sh
```

 **Note:**

You must avoid including the password in the script.

3. Connect to the Administration Server by using the following WLST command:

```
connect('admin_user','admin_password','admin_url')
```

For example:

```
connect('weblogic','<password>','t3s://ADMINVHN:9002')
```

4. Use the `nmEnroll` command to enable the Node Manager to manage servers in a specified WebLogic domain.

```
nmEnroll('ASERVER_HOME')
```

For example:

```
nmEnroll('/u01/oracle/config/domains/soaedg_domain')
```

5. Generate startup properties for the Admin Server using the following WLST command:

```
nmGenBootStartupProps('AdminServer')
```

The `startup.properties` and `boot.properties` files are created in the following directory:

```
$ASERVER_HOME/servers/AdminServer/data/nodemanager/
```

Adding Truststore Configuration to Node Manager

It is required to add the corresponding truststore configuration for Node Manager communication with the different WebLogic Server listeners. To do this, edit Node Manager's start script `startNodeManager.sh` located at `$NM_HOME` and add the variable `JAVA_OPTIONS` to set the `javax.net.ssl.trustStore` and `javax.net.ssl.trustStorePassword` properties with the values used by your EDG system. For example:

```
export JAVA_OPTIONS="${JAVA_OPTIONS} -Djavax.net.ssl.trustStore=/u01/oracle/
config/keystores/appTrustKeyStore.pkcs12 -
Djavax.net.ssl.trustStorePassword=mypassword"
```

 **Note:**

If you have used the `generate_perdomainCACERTS.sh` script to generate certificates and stores, the `trustStorePassword` is the password provided as "KEYPASS" parameter to the script.

Configuring the Domain Directories and Starting the Servers on SOAHOST1

After the domain is created and the node manager is configured, you can then configure the additional domain directories and start the Administration Server and the Managed Servers on SOAHOST1.

- [Starting the Administration Server Using the Node Manager](#)
After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.
- [Validating the Administration Server](#)
Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Remote 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 SOAHOST1](#)
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 SOAHOST1 and SOAHOST2. The domain directory on the local (or private) storage is used to run the Managed Servers.
- [Starting and Validating the WLS_WSM1 Managed Server on SOAHOST1](#)
After you have configured Node Manager and created the Managed Server domain directory, you can use Oracle Enterprise Manager Fusion Middleware Control to start the WLS_WSM1 Managed Server on SOAHOST1.

Starting the Administration Server Using the Node Manager

After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.

To start the Administration Server by using the Node Manager:

1. Ensure that the Administration Server is stopped.
2. Start the WebLogic Scripting Tool (WLST):

```
export WLST_PROPERTIES=""
-Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/
```

```
keystores/appTrustKeyStore.pkcs12
-Dweblogic.security.CustomTrustKeyStorePassPhrase=password"
cd $ORACLE_COMMON_HOME/common/bin
./wlst.sh
```

 **Note:**

The `weblogic.security.SSL.ignoreHostnameVerification=true` is required when using Demo certificates as the ones provided by the `generateCertificates` scripts. In an environment with formal CA and certificates, this flag should not be used.

3. Connect to Node Manager by using the Node Manager credentials:

```
nmConnect('nodemanager_username','nodemanager_password','ADMINVHN','5556',
domain_name','ASERVER_HOME','SSL')
```

Replace `ADMINVHN` and `ASERVER_HOME` with the values of the respective variables.

 **Note:**

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

```
ASERVER_HOME/config/nodemanager
```

4. Start the Administration Server:

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

 **Note:**

When you start the Administration Server, it attempts to connect to Oracle Web Services Manager for WebServices policies. It is expected that the WSM-PM Managed Servers are not yet started, and so, the following message appears in the Administration Server log:

```
<Warning><oracle.wsm.resources.policymanager>
<WSM-02141><Unable to connect to the policy access service due to
Oracle WSM policy manager host server being down.>
```

5. 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 Remote 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:

```
https://ADMINVHN:9002/em
```

You should be able to connect to the Admin Server from the Remote Console as before.

Creating a Separate Domain Directory for Managed Servers on SOAHOST1

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 SOAHOST1 and SOAHOST2. 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 SOAHOST1 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/create_domain.jar \
  -template_name=create_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 `create_domain.jar` file is a sample name for the jar file that you create, which contains the domain configuration files.

- The `create_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 `create_domain.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 SOAHOST1 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/create_domain.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/create_domain.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack 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 SOAHOST1 local storage device.

Starting and Validating the WLS_WSM1 Managed Server on SOAHOST1

After you have configured Node Manager and created the Managed Server domain directory, you can use Oracle Enterprise Manager Fusion Middleware Control to start the WLS_WSM1 Managed Server on SOAHOST1.

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

```
https://ADMINVHN:9002/em
```

In this example:

- Replace *ADMINVHN* with the host name assigned to the ADMINVHN Virtual IP address in [Identifying and Obtaining Software Downloads for an Enterprise Deployment](#).
- Port 9002 is the Administration port used for 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 by using Oracle Enterprise Manager Fusion Middleware, see Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in *Administering Oracle Fusion Middleware*.

2. Sign-in to the Fusion Middleware Control by using the administrator's account.
3. Select the **Servers** pane to view the Managed Servers in the domain.
4. Select only the **WLS_WSM1** Managed Server, and note the assigned port number.
5. Click **Control > Start** on the tool bar to start the selected **WLS_WSM1** Managed Server.
6. To verify that the Managed Server is working correctly, open your browser and enter the following URL:

```
https://SOAHOST1:7010/wsm-pm/
```

Enter the domain admin user name and password when prompted.

 **Note:**

The wsm-pm validator does not work properly until the appropriate WSM Domain Configuration and WSM bootstrap steps are completed as described in the following sections.

Configuring Web Services Manager

This section describes how to configure Web Services Manager.

- [Updating WebServices Domain Configuration](#)
- [Bootstrapping WSM](#)

Updating WebServices Domain Configuration

1. Log into the Fusion Middleware Control by using the administrator's account.
2. In the **Weblogic Domain** drop down menu, select **WebServices > WSM Domain Configuration**.
3. Click **Policies Access** tab.
4. Select the **Auto Discover** and **Use SSL Only** check boxes.
5. In the **SSL Setup** section, select **Oneway**.
6. In KeyStore Type, select JKS (Java Key Store).

 **Note:**

You must select JKS if you are using the certificates and stores created in previous steps.

7. In the Truststore Path enter the location of the truststore used in previous sections as follows:

```
/u01/oracle/config/keystores/appTrustKeyStore.pkcs12
```
8. In the **Key** field, enter a name to uniquely identify the password used for the truststore.
9. In the **password** field, enter the password used for the truststore in previous sections (same as domain admin).
10. Click **Apply**.

Bootstrapping WSM

In a new terminal window, perform the following steps to bootstrap WSMPM.

 **Note:**

If this task is not performed, the WSMPM does not work properly .

1. Change directory to the following directory as follows:

```
cd $ORACLE_COMMON_HOME/common/bin
```

2. Start the WebLogic Server Scripting Tool (WLST). To use the certificates created for the appropriate SSL handshake, the location of the stores and password of the same need to be provided to WLST. Use the following command or add these in a script that can be easily invoked (avoid including the password in the script):

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/
keystores/appTrustKeyStore.pkcs12
-Dweblogic.security.CustomTrustKeyStorePassPhrase=storepassword"
./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','<password>','t3s://ADMINVHN:9002')
```

4. Use the `setWSMBootstrapConfig` to enable WSM `pm.url`.

```
setWSMBootstrapConfig('domain_name','domainpath','ConfigManager','pm.url','
auto-ssl')
```

For example:

```
setWSMBootstrapConfig('soaedg_domain','/u01/oracle/config/domains/
soaedg_domain','ConfigManager','pm.url','auto-ssl')
```

Check that the appropriate entry is created in the `$ASERVER_HOME/config/fmwconfig/wsm-config.xml` for the domain. For example:

```
cat /u01/oracle/config/domains/soaedg_domain/config/fmwconfig/wsm-
config.xml
<?xml version="1.0" encoding="UTF-8"?>
<orares:properties xmlns:orares="http://xmlns.oracle.com/wsm/resources"
xmlns:wsp15="http://www.w3.org/ns/ws-policy" xmlns:orawsp="http://
schemas.oracle.com/ws/2006/01/policy" orares:type="CONFIGURATION"
orares:resource="">
  <orares:property orares:category="ConfigManager" orares:name="pm.url">
    <orares:value>auto-ssl</orares:value>
  </orares:property>
</orares:properties>
```

Propagating the Domain and Starting the Servers on SOAHOST2

After you start and validate the Administration Server and WLS_WSM1 Managed Server on SOAHOST1, you can then perform the following tasks on SOAHOST2.

- [Unpacking the Domain on SOAHOST2](#)
- [Starting the Node Manager on SOAHOST2](#)
- [Starting and Validating the WLS_WSM2 Managed Server on SOAHOST2](#)

Unpacking the Domain on SOAHOST2

This procedure assumes you have copied the file that you created earlier in a location that is accessible from both SOAHOST1 and SOAHOST2; such as the *ASERVER_HOME* directory, which is located on the shared storage filer:

1. Log in to SOAHOST2.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the SOAHOST2 storage device.

Use the examples in [File System and Directory Variables Used in This Guide](#) as a guide.

3. Make sure the `create_domain.jar` accessible to SOAHOST2.

For example, if you are using a separate shared storage volume or partition for SOAHOST2, then copy the template to the volume or partition mounted to SOAHOST2.

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/create_domain.jar \
            -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME
```

 **Note:**

The `-overwrite_domain` option in the `unpack` command allows unpacking a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this `unpack` operation.

Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional java command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the `pack` and `unpack` commands.

In this example:

- Replace `MSERVER_HOME` with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
- Replace `/full_path/create_domain.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device.
- Replace `APPLICATION_HOME` with the complete path to the Application directory for the domain on shared storage. See [File System and Directory Variables Used in This Guide](#).

 **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 SOAHOST2 local storage device.

Starting the Node Manager on SOAHOST2

After you manually set up the Node Manager to use a per host Node Manager configuration, you can start the Node Manager by using the following commands on SOAHOST2:

1. Change directory to the Node Manager home directory:

```
cd $NM_HOME
```

2. Run the following command to start the Node Manager and send the output of the command to an output file, rather than to the current terminal shell:

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

Starting and Validating the WLS_WSM2 Managed Server on SOAHOST2

Use the procedure in [Starting and Validating the WLS_WSM1 Managed Server on SOAHOST1](#) to start and validate the WLS_WSM2 Managed Server on SOAHOST2.

Modifying the Upload and Stage Directories to an Absolute Path

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. See [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).

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 Remote 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](#)

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 SOA Suite](#)
- [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 SOA Suite enterprise deployment domain, then create a user called `weblogic_soa` and an administration group called `SOA Administrators`.

About the Domain Connector User

Oracle recommends that you create a separate domain connector user (for example, `soaLDAP`) 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 Remote Console. Instead, you must use the appropriate LDAP modification tools, such as Idapbrowser 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. Browse to the Fusion Middleware Control, and log in with the administrative credentials.

`https://ADMINVHN:9002/em`

2. Navigate to **WebLogic Domain > Security > Security Provider Configuration**.
3. Expand **Security Store Provider**.
4. Expand **Identity Store Provider**.
5. Click **Configure**.
6. Add a custom property.
7. Set the following properties:

- `virtualize` with value `true`
- `optimize_search` with value `true`

Click **OK**.

8. Click **OK** again to persist the change.
9. Restart the Administration Server and all managed servers.

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

 **Note:**

When you set `virtualize` to `true`, applications that create users or groups in the LDAP require two additional custom properties in the Identity Store Provider:

- Property `user.create.bases`, to specify the DN under which the users will be created. Example: `cn=users,dc=example,dc=com`.
- Property `group.create.bases`, to specify the DN under which the groups will be created. Example: `cn=groups,dc=example,dc=com`.

You can configure these properties by following the steps that are described above for adding the `virtualize` property.

SOA products do not have any application that creates users or groups in the LDAP, so this is required only if you are planning to deploy any additional application that does it. See *Configuring the Identity Store in [Securing Applications with Oracle Platform Security Services](#)*

About Product-Specific Roles and Groups for Oracle SOA Suite

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 `SOA Administrators` group. After they are added to the `SOA Administrators` 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 `SOA Administrators` 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_soa,cn=users,dc=example,dc=com`
- Admin Group DN:
`cn=SOA Administrators,cn=groups,dc=example,dc=com`
- Product-specific LDAP Connector User:
`cn=soaLDAP,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 `soaLDAP` 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=soaLDAP,cn=systemids,dc=example,dc=com
changetype: add
orclsamaccountname: soaLDAP
userpassword: password
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetorgperson
objectclass: orcluser
objectclass: orcluserV2
mail: soaLDAP@example.com
givenname: soaLDAP
sn: soaLDAP
cn: soaLDAP
uid: soaLDAP
```

 **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=soaLDAP,cn=systemids,dc=example,dc=com
```

```
dn: cn=orclFAGroupReadPrivilegeGroup,cn=groups,dc=example,dc=com
changetype: modify
add: uniquemember
uniquemember: cn=soaLDAP,cn=systemids,dc=example,dc=com
```

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

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

2. Provision the user in the LDAP directory.

For example, for an Oracle Unified Directory LDAP provider:

```
OID_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 into the WebLogic Remote Console.
2. Click **Edit Tree**.

- In the left navigational bar, click **Security > Realms > myrealm > Authentication Providers**.

 **Note:**

The `DefaultAuthenticator` provider is configured for the realm. This is the default WebLogic Server authentication provider.

- Click **New** button.
- Enter a name for the provider.

Use one of the following names, based on the LDAP directory service that you plan to use as your credential store:

- `OUAuthenticator` for Oracle Unified Directory
- `OIDAuthenticator` for Oracle Internet Directory

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

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

- `OracleUnifiedDirectoryAuthenticator` for Oracle Unified Directory
- `OracleInternetDirectoryAuthenticator` for Oracle Internet Directory

- Select **SUFFICIENT** from the **Control Flag** drop-down menu for the newly created authenticator provider.

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 falls 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` option and make sure that its control flag is set to **SUFFICIENT**.

- Click **Create**.
- Click the **Authenticator Parameters** 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: <code>idstore.example.com</code>	The LDAP server's server ID.
Port	For example: <code>1389</code>	The LDAP server's port number.
Principal	For example: <code>cn=soaLDAP, cn=systemids, dc=example, dc=com</code>	The LDAP user DN used to connect to the LDAP server.

Parameter	Sample Value	Value Description
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))	<p>Instead of a default search criteria for All Users Filter, search all users based on the <code>uid</code> value.</p> <p>If the User Name Attribute for the user object class in the LDAP directory structure is a type other than <code>uid</code>, then change that type in the User From Name Filter field.</p> <p>For example, if the User Name Attribute type is <code>cn</code>, then this field should be set to: (&(cn=*) (objectclass=person))</p>
User From Name Filter	For example: (&(uid=%u) (objectclass=person))	<p>If the User Name Attribute for the user object class in the LDAP directory structure is a type other than <code>uid</code>, then change that type in the settings for the User From Name Filter.</p> <p>For example, if the User Name Attribute type is <code>cn</code>, then this field should be set to: (&(cn=%u) (objectclass=person))</p>
User Name Attribute	For example: uid	The attribute of an LDAP user object that specifies the name of the user.
Group Base DN	For example: cn=groups,dc=example,dc=com	Specify the DN that points to your Groups node.
Use Retrieved User Name as Principal	Checked	Must be turned on.
GUID Attribute	entryuuid	This value is prepopulated with <code>entryuuid</code> when <code>OracleUnifiedDirectoryAuthenticator</code> is used for OUD. Check this value if you use Oracle Unified Directory as your authentication provider.

10. Click **Save** to save the changes.
11. Commit the changes in the shopping cart.
12. Navigate to **Authenticator Providers** under **Security > Realms > myrealm**.
13. Check the Authenticator Providers you just created and move up to the first position.
14. On the Authentication Providers screen, click **DefaultAuthenticator**.
15. From the Control Flag drop-down, select **SUFFICIENT**.
16. Click **Save** to update the DefaultAuthenticator settings.
17. Commit the changes in the shopping cart.
18. Restart the Administration Server and all managed servers.

To stop the Managed Servers, sign 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 by using the Node Manager:

a. Start WLST:

```
export WLST_PROPERTIES="
-Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/keystores/
appTrustKeyStore.pkcs12
-Dweblogic.security.CustomTrustKeyStorePassPhrase=password"
cd $ORACLE_COMMON_HOME/common/bin
./wlst.sh
```

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

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

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.

 **Note:**

If you plan to log in to the system immediately by using the central LDAP user role, you can skip the restart until you have assigned the Administration role to the new enterprise deployment administration group. For more information, see [Adding the Administration Role to the New Administration Group](#).

19. 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 "OUDatauthenticator" 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 that they are correct and try saving and restarting the Administration Server again.

20. 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:

- a. In the Remote Console, go to the **Security Tree**.
- b. Navigate to **Realms > myrealm > Authentication Providers**.
- c. Expand the new Authentication Provider.

- d. Click **Users** and then click **Groups**.

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_soa` and a group called `SOA Administrators`.

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

2. Provision the user in the LDAP directory.

For example, for an Oracle Unified Directory LDAP provider:

```
OID_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=oudadmin" \
                              -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=SOA Administrators,cn=Groups,dc=example,dc=com
displayname: SOA Administrators
objectclass: top
objectclass: GroupOfUniqueNames
```



```

objectclass: orclGroup
uniquemember: cn=weblogic_soa,cn=users,dc=example,dc=com
cn:SOA Administrators
description: Administrators Group for the Oracle SOA Suite 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=oudadmin" \
-w password \
-c \
-v \
-f admin_group.ldif

```

5. Verify that the changes were made successfully:
 - a. In the Remote Console, go to **Security Tree**.
 - b. Navigate to **Realms > myrealm > Authentication Providers**.
 - c. Expand the new Authentication Provider.
 - d. Click **Users** and verify if the administrator user that you provisioned is listed.
 - e. Click **Groups** and verify if the administrator group that you provisioned is listed.

Adding the Administration Role to the New Administration Group

After you add the users and groups to your LDAP 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 Remote 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. Click **Security Data Tree**.
3. Click **Realms > myrealm > Role Mappers > XACMLRoleMapper > Global > Roles**.
4. Click the **Admin** role.
5. Click **Add conditions**.
6. Select **Group** from the **Predicate List** drop-down menu, and then click **Next**.
7. Enter `SOA Administrators` in the **Group Argument Name** field, and then click **OK**.

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

8. Click **Save** to finish adding the **Admin** Role to the `SOA Administrators` group.
9. Validate that the changes were made by logging into the WebLogic Remote Console and into the Fusion Middleware Control by using the new `weblogic_soa` user credentials.

If you can log into the Oracle WebLogic Remote 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 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 (`SOA Administrators`) as a member to the `policy.Updater` role in the `wsm-pm` application stripe.

1. Sign in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_soa`.
2. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.
3. Select the **wsm-pm** application stripe from the Application Stripe drop-down menu.
4. Click the triangular icon next to the role name text box to search for all role names in the `wsm-pm` application stripe.
5. Select the row for the **policy.Updater** role to be edited.
6. Click the Application Role **Edit** icon to edit the role.
7. Click the Application Role **Add** icon on the Edit Application Role page.
8. In the Add Principal dialog box, select **Group** from the **Type** drop-down menu.
9. To search for the enterprise deployment administrators group, enter the group name `SOA Administrators` in the **Principal Name Starts With** field and click the right arrow to start the search.
10. Select the appropriate administrators group in the search results and click **OK**.
11. Click **OK** on the Edit Application Role page.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see [Performing Backups and Recoveries for an Enterprise Deployment](#).

Verification of Manual Failover of the Administration Server

After you configure the domain, you should test the failover. For instructions to test, see [Verifying Manual Failover of the Administration Server](#).

Configuring Oracle HTTP Server for an Enterprise Deployment

For an enterprise deployment, Oracle HTTP Server must be installed on each of the web tier hosts and configured as Oracle HTTP standalone domains on each host.

In this enterprise deployment, the LBR communicates with OHS over SSL protocol for a more secure configuration. The OHS instances also communicate over SSL protocol with the specific Managed Servers in the application tier. SSL is configured all the way from the LBR to the backend WLS servers.

Before you configure Oracle HTTP Server, be sure to review [Understanding the Web Tier](#).

 **Note:**

As of Fusion Middleware 14.1.2.0.0, Oracle Traffic Director has been deprecated. For an enterprise deployment, use Oracle HTTP Server.

- [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 simplified management that requires a minimum amount of configuration and a minimum amount of resources to run and maintain. Contrary to the App tier, Node Managers in the Web Tier listen on plain sockets because they are only accessed locally (they listen on localhost only).
- [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.
- [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.
- [Generate Required Certificates for OHS SSL Listeners](#)
Since the OHS listeners use SSL it is necessary to create the appropriate certificates for them and add also the pertaining SANs for the server names they use. It is required to have certificates for each WEBHOST address, adding as SAN the different ServerNames that are used in them.

- [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 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 simplified management that requires a minimum amount of configuration and a minimum amount of resources to run and maintain. Contrary to the App tier, Node Managers in the Web Tier listen on plain sockets because they are only accessed locally (they listen on localhost only).

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

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`
- `WEB_KEYSTORE_HOME`
- `JAVA_HOME`

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

- `ADMINVHN`
- `WEBHOST1`
- `WEBHOST2`
- `SOAHOST1`
- `SOAHOST2`

Installing Oracle HTTP Server on WEBHOST1

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

- [Installing a Supported JDK](#)
- [Starting the Installer on WEBHOST1](#)
- [Navigating the Oracle HTTP Server Installation Screens](#)
- [Verifying the Oracle HTTP Server 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](#)

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:

<https://www.oracle.com/java/technologies/downloads/>

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

Installing the JDK Software

Oracle HTTP Server requires that you install a certified Java Development Kit (JDK) on your system.

You must install the JDK in the local storage device for each of the web tier host computers. The web tier host computers, which reside in the DMZ, do not necessarily have access to the shared storage on the application tier.

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

The following example describes how to install a recent version of JDK 17.0.10.

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 -xzf jdk-17.0.10+11_linux-x64_bin.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 ./jdk-17.0.10 /u02/oracle/products/jdk
```

See [File System and Directory Variables Used in This Guide](#).

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

For example:

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

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

```
java -version
```

The Java version in the output should be displayed as `17.0.10`.

6. Repeat steps 1 through 5 for each web tier host. For example, `WEBHOST1` and `WEBHOST2`.

Starting the Installer on WEBHOST1

To start the installation program, perform the following steps.

1. Log in to `WEBHOST1`.
2. Go to the directory in which you downloaded the installation program.
3. Enter the following command to launch the installation program:

```
./fmw_14.1.2.0.0_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	<p>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.</p> <p>See Understanding the Oracle Central Inventory in <i>Installing Software with the Oracle Universal Installer</i>.</p>
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Oracle recommends that you configure the central inventory directory within the products directory. Example: <code>/u02/oracle/products/oraInventory</code></p> <p>You may also need to execute the <code>createCentralInventory.sh</code> script as root from the <code>oraInventory</code> folder after the installer completes.</p> </div>
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	<p>Use this screen to specify the location of your Oracle home directory.</p> <p>For the purposes of an enterprise deployment, enter the value of the <code>WEB_ORACLE_HOME</code> variable listed in Table 7-3.</p>
Installation Type	<p>Select Standalone HTTP Server (Managed independently of WebLogic server).</p> <p>This installation type allows you to configure the Oracle HTTP Server instances independently from any other existing Oracle WebLogic Server domains.</p>
JDK Selection	For the value of JDK Home, enter the value of <code>JAVA_HOME</code> that you set when installing the JDK software.
Prerequisite Checks	<p>This screen verifies that your system meets the minimum necessary requirements.</p> <p>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.</p>

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

Screen	Description
Installation Summary	Use this screen to verify the installation options that you selected. If you want to save these options to a response file, click Save Response File and provide the location and name of the response file. Response files can be used later in a silent installation situation. See Using the Oracle Universal Installer in Silent Mode in <i>Installing Software with the Oracle Universal Installer</i> .
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click Finish to close the installer.

Verifying the Oracle HTTP Server Installation

Verify that the Oracle HTTP Server installation completed successfully by validating the `WEB_ORACLE_HOME` folder contents.

Run the following command to compare the installed folder structure with the following list:

```
ls --format=single-column $WEB_ORACLE_HOME
```

The following files and directories are listed in the Oracle HTTP Server Oracle Home:

```
assistants  
bin  
cfgtoollogs  
clone  
crs  
crypto  
css  
cv  
deinstall  
drdaas  
env.ora  
has  
hs  
install  
instantclient  
inventory  
javavm  
jdbc  
jlib  
jpub  
ldap  
lib  
network  
nls  
odbc  
ohs
```

```
olap
OPatch
opmn
oracle_common
oracore
oraInst.loc
ord
oss
oui
perl
plssql
plugins
precomp
QOpatch
racg
rdbms
root.sh
schagent.conf
sdk
slax
sqlcl
sqlj
sqlplus
srvm
suptools
ucp
unixODBC
usm
utl
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.

Tip:

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

Task 2 Selecting the Configuration Templates

On the Templates screen, select **Oracle HTTP Server (Standalone) - [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. 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 the value of `WEBHOST1`.

All the remaining fields are prepopulated, but you can change the values as required for your organization. The non-ssl listener will be disabled manually later in this guide. 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](#)
- [Setting Frontend Addresses and WebLogic Plugin for the WSM_PM Cluster and the Administration Server](#)

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:

```
cd $WEB_DOMAIN_HOME/nodemanager
```

2. Modify `nodemanager.properties` to not use secure listener. Ensure it uses the localhost only as listen address. Your `$WEB_DOMAIN_HOME/nodemanager/nodemanager.properties` should appear like the following:

```
#Mon Feb 26 18:12:34 GMT 2024
#Node manager properties
#Mon Feb 26 18:03:35 GMT 2024
LogAppend=true
DomainsFile=/u02/oracle/config/domains/soaedgohs/nodemanager/
nodemanager.domains
LogLevel=INFO
PropertiesVersion=14.1.2.0.0
ListenBacklog=50
QuitEnabled=false
LogCount=1
LogLimit=0
NodeManagerHome=/u02/oracle/config/domains/soaedgohs/nodemanager
LogToStderr=true
NativeVersionEnabled=true
AuthenticationEnabled=true
CrashRecoveryEnabled=false
weblogic.StopScriptEnabled=false
DomainsFileEnabled=true
weblogic.StartScriptEnabled=true
LogFile=/u02/oracle/config/domains/soaedgohs/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenAddress=localhost
JavaHome=/u02/oracle/products/jdk
weblogic.StartScriptName=startWebLogic.sh
ListenPort=5556
SecureListener=false
StateCheckInterval=500
```

3. 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 &
```

4. 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. To start the ohs1 instance in WEBHOST1, enter the following commands:

```
$WEB_ORACLE_HOME/oracle_common/common/bin/wlst.sh

wls:/offline>
nmConnect('ohsdomain_admin_user','ohsdomain_admin_password','localhost','55
56','ohsdomain_name','WEB_DOMAIN_HOME','PLAIN')

wls:/nm/soaedgohs> nmStart(serverName='ohs1',serverType='OHS')
```

2. Repeat *Step 1* to start the ohs2 instance on WEBHOST2. See Starting Oracle HTTP Server Instances in *Administering Oracle HTTP Server*.
3. Check the logs in each node at \$WEB_DOMAIN_HOME/servers/ohs1/logs/ohs1.log.

This will allow you to validate the appropriate start of OHS on a non-ssl listener. The following steps will guide you through the process for using the SSL listeners for OHS and routing to WLS using SSL.

Setting Frontend Addresses and WebLogic Plugin for the WSM_PM Cluster and the Administration Server

As a security best practice Oracle recommends setting a frontend address for the Administration Server and the WSM-PM cluster. In the initial domain creation steps, since OHS and the frontend Load Balancer may have not been configured yet, the frontend setting is avoided to allow verifications using the individual server addresses. However, at this point and before configuring OHS (and the frontend load balancer, if not done yet) it is required to add the pertaining addresses.

1. To set the frontend and WebLogic Plugin for the Administration Server, use the WebLogic Remote Console as follows:
 - a. Click **Edit Tree**.
 - b. Click **Environment>Servers>AdminServer**.
 - c. Select the **Protocol** Tab and then select the **HTTP** tab.
 - d. As **Frontend Host**, enter the front end LBR address that is used to access Enterprise management and the Remote Console (*admin.example.com* in the example used in this guide).
 - e. Leave the **Frontend HTTP** port set to 0.
 - f. Enter the LBR's admin listener port (445) as **Frontend HTTPS port**.
 - g. Click **Save**.
 - h. Click the cart icon at the top right to commit the changes.
2. To set the frontend for the WSM-PM Cluster, use the Remote Console as follows:
 - a. Click **Edit Tree**.
 - b. Click **Environment>Clusters>WSM-PM_Cluster**.

- c. Select the **HTTP** tab.
 - d. As **Frontend Host**, enter the front end LBR address that will be used to access internally to the Enterprise Deployment system (services like WSMPPM and *internal.example.com* in the example used in this guide).
 - e. Leave the **Frontend HTTP** port set to 0.
 - f. Enter the LBR's internal listener (if your LBR does not allow using the same port from different listener this will have to be a different one from the admin port used for the Admin Server access) as **Frontend HTTPS port** (444).
 - g. Click **Save**.
 - h. Click the cart icon at the top right to commit the changes.

These changes requires a restart of the AdminServer and the `WSM-PM_Cluster` to be effective (a notification appears in the WebLogic Remote Console about restart being required).
3. Enable the proxy plugin for the domain using the WebLogic Remote Console as follows:
 - a. Click **Edit Tree**.
 - b. Click **Environment>Domain**.
 - c. Select **Web Application** tab.
 - d. Click the **WebLogic Plugin Enable** button.
 - e. Click **Save**.
 - f. Click the cart icon at the top right to commit the changes.

Generate Required Certificates for OHS SSL Listeners

Since the OHS listeners use SSL it is necessary to create the appropriate certificates for them and add also the pertaining SANs for the server names they use. It is required to have certificates for each WEBHOST address, adding as SAN the different ServerNames that are used in them.

This enterprise deployment uses *soainternal.example.com*, *soa.example.com*, *osb.example.com* and *admin.example.com* as frontend addresses. These addresses are used in the WLS domain configuration as frontend addresses for different clusters and servers.

Oracle recommends using the same Identity and Trust store files for all the CAs and certificates used in the app tier. The OHS nodes, do not use shared storage so the stores need to be copied to their private folders from the app tier. Certificates in a production system should come from formal Certificate Authorities.

In Oracle FMW 14.1.2.0, the Oracle WebLogic allows the usage of a per-domain Certificate Authority (CA). To update the Identity store and a Trust Store for the OHS SSL listeners in a Weblogic Server using a per-domain CA, you can perform the following steps. Run these steps in any of the WLS nodes (because the OHS ones do not install the CerGen and keytool utilities) and then transfer the stores to the OHS nodes:

1. Download the `generate_perdomainCACERTS-ohs.sh` script from the maa github repo https://github.com/oracle-samples/maa/blob/main/1412EDG/generate_perdomainCACERTS-ohs.sh to SOAHOST1.
2. Run the script with the following arguments:
 - `WLS_DOMAIN_DIRECTORY`: Directory hosting the Weblogic Domain that the Administration Server uses (ASERVER variable in this guide).

- **WL_HOME:** The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored. Typically `/u01/oracle/products/fmw/wlserver`.
- **KEYSTORE_HOME:** Directory where the `appIdentity` and the `appTrust` stores are created.
- **KEYPASS:** Password used for the weblogic administration user (reused for certs and stores).
- **LIST_OF_OHS_SSL_VIRTUAL_HOSTS:** A space separated list of OHS Virtual host addresses enclosed in single quotes.

For example:

```
./generate_perdomainCACERTS-ohs.sh /u01/oracle/config/domains/soaedg_domain /u01/oracle/products/fmw/wlserver /u01/oracle/config/keystores "password" 'ohshost1.example.com ohshost2.example.com'
```

The script performs the following actions:

- a. It traverses the domain configuration and extracts the front-end addresses used by the domain.
- b. It uses the per domain CA to generate certificates for the OHS addresses that are provided as input to the script. The front-end addresses gathered from the domain configuration are added as SAN Subject Alternative Names (SAN) to these certificates.
- c. It connects to the front-end addresses detected in the domain configuration and downloads their public certificates. It adds these certificates to the WebLogic's trust keystore to allow the WebLogic servers establish SSL handshake with the different front-end addresses (used for callbacks, identity access and other redirections).

 **Note:**

The node where the `generate_perdomainCACERTS-ohs.sh` script is executed needs to have connectivity to the different front-end addresses included in the domain's `config.xml` to download their certificates.

- d. It uses `orapki` to convert the identity and trust stores in the application tier into the required pkcs wallets used by the different OHS Virtual Hosts.
- e. It creates a tar with the corresponding wallets (this needs to be transferred to the OHS nodes for completing the SSL configuration).
3. Transfer the tar generated by the script to the OHS nodes.
 - a. Use `scp` or any `sftp` tool to copy tar file to the OHS nodes. For consistency with the app tier, place it under `$WEB_KEYSTORE_HOME`.
 - b. Untar the contents of the file in that folder as follows:
 - i. `cd $WEB_KEYSTORE_HOME`
 - ii. `tar -xzf orapki-ohs.tgz`

This creates a wallet for WLS access and a directory wallet for each virtual host provided as parameter.

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](#)
- [Validating the Virtual Server Configuration on the Load Balancer](#)
- [Validating Access to the Management Consoles and Administration Server](#)
- [Configure a New Provider in the WebLogic Remote Console to Access the Domain Configuration Through the Frontend LBR](#)

About the Oracle HTTP Server Configuration for an Enterprise Deployment

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

- [Purpose of the Oracle HTTP Server Virtual Hosts](#)
- [About the WebLogicCluster Parameter of the <VirtualHost> Directive](#)
- [Recommended Structure of the Oracle HTTP Server Configuration Files](#)

Purpose of the Oracle HTTP Server Virtual Hosts

The reference topologies in this guide require that you define a set of virtual servers on the hardware load balancer. You can then configure Oracle HTTP Server to recognize requests to specific virtual hosts (that map to the load balancer virtual servers) by adding `<VirtualHost>` directives to the Oracle HTTP Server instance configuration files.

For each Oracle HTTP Server virtual host, you define a set of specific URLs (or context strings) that route requests from the load balancer through the Oracle HTTP Server instances to the appropriate Administration Server or Managed Server in the Oracle WebLogic Server domain.

About the WebLogicCluster Parameter of the <VirtualHost> Directive

A key parameter of the Oracle HTTP Server `<VirtualHost>` directive is the `WebLogicCluster` parameter, which is part of the WebLogic Proxy Plug-In for Oracle HTTP Server. When you configure Oracle HTTP Server for an enterprise deployment, consider the following information when you add this parameter to the Oracle HTTP Server configuration files.

The servers specified in the `WebLogicCluster` parameter are important only at startup time for the plug-in. The list needs to provide at least one running cluster member for the plug-in to discover other members of the cluster. When you start the Oracle HTTP server, the listed cluster member must be running. Oracle WebLogic Server and the plug-in work together to update the server list automatically with new, failed, and recovered cluster members.

Some example scenarios:

- Example 1: If you have a two-node cluster and then add a third member, you do not need to update the configuration to add the third member. The third member is discovered on the fly at runtime.
- Example 2: You have a three-node cluster but only two nodes are listed in the configuration. However, if both listed nodes are down when you start Oracle HTTP Server, then the plug-in would fail to route to the cluster. You must ensure that at least one of the listed nodes is running when you start Oracle HTTP Server.

If you list all members of the cluster, then you guarantee you can route to the cluster, assuming at least one member is running when Oracle HTTP Server is started.

Recommended Structure of the Oracle HTTP Server Configuration Files

Rather than adding multiple virtual host definitions to the `httpd.conf` file, Oracle recommends that you create separate, smaller, and more specific configuration files for each of the virtual servers required for the products that you are deploying. This avoids populating an already large `httpd.conf` file with additional content, and it can make troubleshooting configuration problems easier.

For example, in a typical Oracle Fusion Middleware Infrastructure domain, you can add a specific configuration file called `admin_vh.conf` that contains the virtual host definition for the Administration Server virtual host (ADMINVHN).

Since all virtual hosts in this EDG use SSL, the original `ssl.conf` file is used as a template for them. This Enterprise Deployment Guide segregates the listeners and certificates that are used by the different endpoints exposed through OHS. It uses different certificates and listeners for the external, internal and administration virtual hosts. This permits segregating the traffic and the encryption quality for each type of access and provides a well-structured mapping of front ends, Virtual Hosts and listeners.

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. Ensure `ssl.conf` is included in the `httpd` configuration.

```
grep ssl.conf httpd.conf
include "ssl.conf"
```

5. Copy the `ssl.conf` file to a different file name.

 **Note:**

This is used as a template for other module conf files.

```
cp $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
ssl.conf $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
ssl.template
```

6. Edit the `ssl.conf` file to include only the following lines (remove other content from the file):

```
<IfModule ossl_module>
#
# Some MIME-types for downloading Certificates and CRLs
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl

# Inter-Process Session Cache:
# Configure the SSL Session Cache: First the mechanism
# to use, second the expiring timeout (in seconds) and third
# the mutex to be used.
```

```

        SSLSessionCache "shmcb:${ORACLE_INSTANCE}/servers/${COMPONENT_NAME}/
logs/ssl_scache(512000) "
        SSLSessionCacheTimeout 300

</IfModule>

```

7. Modify the `$WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/mod_wl_ohs.conf` to include the appropriate `WLSWallet` file (required to route on SSL to the WLS backends) as follows:

```

# NOTE : This is a template to configure mod_weblogic.

LoadModule weblogic_module    "${PRODUCT_HOME}/modules/mod_wl_ohs.so"

# This empty block is needed to save mod_wl related configuration from EM
to this file when changes are made at the Base Virtual Host Level
<IfModule weblogic_module>

    WLIOWriteTimeoutSecs 900
    KeepAliveSecs 290
    FileCaching OFF
    WLSocketTimeoutSecs 15
    ErrorPage http://www.oracle.com/splash/cloud/index.html
    WLRetryOnTimeout NONE
    WLForwardUriUnparsed On
    SecureProxy On
    WLSWallet "/u02/oracle/config/keystores/orapki/"
</IfModule>

```

8. Log in to `WEBHOST2` and perform steps from 2 to 7, replacing any occurrences of `WEBHOST1` or `ohs1` with `WEBHOST2` or `ohs2` in the instructions as necessary.

Creating the Virtual Host Configuration Files

To create the virtual host configuration files:



Note:

Before you create the virtual host configuration files, be sure that you have configured the virtual servers on the load balancer, as described in [Purpose of the Oracle HTTP Server Virtual Hosts](#).

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

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

2. Copy the `ssl.template` file to the `admin_vh.conf` file, this will transfer most of the required SSL configuration:

```
cp $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf/
```

```
ssl.template $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/
moduleconf/admin_vh.conf
```

3. Edit the file to add the following Listen, VirtualHost, ServerName, AllowEncodedSlashes, and Location directives. Also, change the SSLWallet directory to point to the specific wallet for the virtual host. The admin_vh.conf file should resemble the following file.

```
#####
# Oracle HTTP Server mod_oss1 configuration file: ssl.conf      #
#####

# The Listen directive below has a comment preceding it that is used
# by tooling which updates the configuration. Do not delete the comment.
#[Listen] OHS_SSL_PORT
Listen WEBHOST1:4445
##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost WEBHOST1:4445>
ServerName admin.example.com:445
AllowEncodedSlashes On
  <IfModule oss1_module>

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # Client Authentication (Type):
    # Client certificate verification type and depth. Types are
    # none, optional and require.
    SSLVerifyClient None

    # SSL Protocol Support:
    # Configure usable SSL/TLS protocol versions.
    SSLProtocol TLSv1.2 TLSv1.3

    # Option to prefer the server's cipher preference order
    SSLHonorCipherOrder on

    # SSL Cipher Suite:
    # List the ciphers that the client is permitted to negotiate.
    SSLCipherSuite
    TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,
    TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SH
    A384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_S
    HA384

    #Path to the wallet
    #SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/$
    {COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/default"
    SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1"

    <FilesMatch "\.(cgi|shtml|phtml|php)$">
      SSLOptions +StdEnvVars
```

```

</FilesMatch>

<Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/${
COMPONENT_TYPE}/instances/${COMPONENT_NAME}/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

# Add the following directive to add HSTS
<IfModule mod_headers.c>
    Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
</IfModule>

<Location /em>
    WLSRequest ON
    WebLogicHost ADMINVHN
    WebLogicPort 9002
</Location>
<Location /management>
    WLSRequest ON
    WebLogicHost ADMINVHN
    WebLogicPort 9002
</Location>

</IfModule>
</VirtualHost>

```

4. Repeat similar steps to create a `soainternal_vh.conf` file with this content (notice the different listen port, virtual host, and WLS settings):

```

#####
# Oracle HTTP Server mod_oss1 configuration file: ssl.conf      #
#####

# The Listen directive below has a comment preceding it that is used
# by tooling which updates the configuration. Do not delete the comment.
#[Listen] OHS_SSL_PORT
Listen WEBHOST1:4444

##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost WEBHOST1:4444>
ServerName soainternal.example.com:444
    <IfModule oss1_module>
        # SSL Engine Switch:
        # Enable/Disable SSL for this virtual host.
        SSLEngine on

```

```

# Client Authentication (Type):
# Client certificate verification type and depth. Types are
# none, optional and require.
SSLVerifyClient None

# SSL Protocol Support:
# Configure usable SSL/TLS protocol versions.
SSLProtocol TLSv1.2 TLSv1.3

# Option to prefer the server's cipher preference order
SSLHonorCipherOrder on

# SSL Cipher Suite:
# List the ciphers that the client is permitted to negotiate.
SSLCipherSuite
TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_S
A384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_S
HA384

#Path to the wallet
#SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/${
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/default"
SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1"

<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>

<Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/${
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

# Add the following directive to add HSTS
<IfModule mod_headers.c>
    Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
</IfModule>

<Location /wsm-pm>
WLSRequest ON
WebLogicCluster SOAHOST1:7010,SOAHOST2:7010
</Location>

</IfModule>
</VirtualHost>

```


- Restart the `ohs1` instance by entering the following commands:

```
$WEB_ORACLE_HOME/oracle_common/common/bin/wlst.sh
```

```
wls:/offline>
```

```
nmConnect('ohsdomainadmin','ohsdomainadmin_password','localhost','5556','ohsdomainname','WEB_DOMAIN_HOME','PLAIN')
```

```
wls:/nm/soaedgohs> nmKill(serverName='ohs1',serverType='OHS')
```

```
wls:/nm/soaedgohs> nmStart(serverName='ohs1',serverType='OHS')
```

Watch the `$WEB_DOMAIN_HOME/servers/ohs1/logs/ohs1.log` file for errors.

- Copy the `admin_vh.conf` file and the `soainternal_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (`ohs2`) on `WEBHOST2`:

```
$WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf
```

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

- Restart the `ohs2` instance by entering the following commands:

```
$WEB_ORACLE_HOME/oracle_common/common/bin/wlst.sh
```

```
wls:/offline>
```

```
nmConnect('ohsdomainadmin','ohsdomainadmin_password','localhost','5556','ohsdomainname','WEB_DOMAIN_HOME','PLAIN')
```

```
wls:/nm/soaedgohs> nmKill(serverName='ohs2',serverType='OHS')
```

```
wls:/nm/soaedgohs> nmStart(serverName='ohs2',serverType='OHS')
```

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 12c web page.

- <https://admin.example.com:445/index.html>
- <https://soainternal.example.com:444/index.html>

Validating Access to the Management Consoles and Administration Server

To verify the changes that you have made in this chapter:

- Access the Fusion Middleware Control by using the following URL:

```
https://admin.example.com:445/em
```

Configure a New Provider in the WebLogic Remote Console to Access the Domain Configuration Through the Frontend LBR

Create a new Admin Server Connection Provider that connects through the frontend load balancer and OHS to the domain's Administration Server. To establish this connection, the WebLogic Remote Console must trust the certificate used by the load balancer for the administration frontend address.

1. Ensure that the Trust Store used by the WebLogic Remote Console includes the certificate or the CA certificate used by the frontend load balancer in the admin virtual server.

 **Tip:**

If you used the script `generate_perdomainCACERTS-ohs.sh`, you can download the `appTrustKeyStore.pkcs12` file from the domain and use it as the WebLogic Remote Console trust store. It includes the frontend load balancer certificates as a trusted entity.

2. Open the WebLogic Remote Console and click **Add Admin Server Connection Provider**.
3. Use the following values for the new provider:
 - **Connection Provider Name:**
Use a name identifying the connection. For example, `soaedg_domain_lbrprovider`.
 - **Username and Password:**
Enter the WebLogic Domain Administration user and password.
 - **URL:** Use the frontend address and the port. For example, `https://admin.example.com:445`.
 - **Make Insecure Connect:** If the the appropriate trust store settings are completed, you do not need to check this field.

 **Note:**

If you are using demo certs in the load balancer, you might need to check the **Disable host name verification** field in the WebLogic Remote Console settings.

4. Click **OK** to add the provider.
5. Click the new provider.

You must be able to manage the domain remotely through the front end LBR with these settings.

Extending the Domain with Oracle SOA Suite

You need to perform certain tasks in order to extend the enterprise deployment domain with the Oracle SOA Suite software.

- [Variables Used When Configuring Oracle SOA Suite](#)
While extending the domain with Oracle SOA Suite, you are referencing the directory variables listed in this section.
- [Synchronizing the System Clocks](#)
Verify that the system clocks on each host computer are synchronized.
- [Installing the Software for an Enterprise Deployment](#)
The procedure to install the software for an enterprise deployment is explained in this section.
- [Creating the Oracle SOA Suite Database Schemas](#)
Before you can configure an Oracle SOA Suite domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.
- [Extending the Enterprise Deployment Domain with Oracle SOA Suite](#)
Perform the following tasks to extend the existing enterprise deployment domain with the Oracle SOA Suite software.
- [Update Certificates for New Frontend Addresses](#)
This section contains information about certificates for new frontend addresses.
- [Update the WebLogic Servers Security Settings](#)
This section contains information about WebLogic Servers security settings.
- [Propagating the Extended Domain to the Domain Directories and Machines](#)
After you have extended the domain with the Oracle SOA Suite instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and machines.
- [Starting and Validating the WLS_SOA1 Managed Server](#)
Now that you have extended the domain, started the Administration Server, and propagated the domain to the other hosts, you can start the newly configured Oracle SOA Suite Managed Servers.
- [Starting and Validating the WLS_SOA2 Managed Server](#)
After you validate the successful configuration and startup of the WLS_SOA1 Managed Server, you can start and validate the WLS_SOA2 Managed Server.
- [Modifying the Upload and Stage Directories to an Absolute Path](#)
- [Configuring the Web Tier for the Extended Domain](#)
Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.
- [Post-Configuration Steps for Oracle SOA Suite](#)
After you install and configure Oracle SOA Suite, consider the following post-configuration tasks.
- [Replacing Connect Strings with the Appropriate TNS Alias](#)
Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

- [Backing Up the Configuration](#)
It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

Variables Used When Configuring Oracle SOA Suite

While extending the domain with Oracle SOA Suite, you are referencing the directory variables listed in this section.

The values for several directory variables are defined in [File System and Directory Variables Used in This Guide](#).

- ORACLE_HOME
- ASERVER_HOME
- MSERVER_HOME
- APPLICATION_HOME
- DEPLOY_PLAN_HOME
- WEB_DOMAIN_HOME
- JAVA_HOME
- ORACLE_RUNTIME

In addition, you reference the following virtual IP (VIP) address that are defined in [Reserving the Required IP Addresses for an Enterprise Deployment](#):

- ADMINVHN

Actions in this chapter are performed on the following host computers:

- SOAHOST1
- SOAHOST2
- WEBHOST1
- WEBHOST2

Synchronizing the System Clocks

Verify that the system clocks on each host computer are synchronized.

Oracle recommends the use of the Network Time Protocol (NTP). See [Configuring a Host to Use an NTP \(time\) Server](#).

To verify the time synchronization, query the NTP service by running the `chronyc -n tracking` command on each host.

Sample output:

```
$chronyc -n tracking
Reference ID : A9FEA9FE (169.254.169.254)
Stratum : 3
Ref time (UTC) : Tue Jan 14 15:28:01 2025
System time : 0.000043127 seconds fast of NTP time
```

Last offset : +0.000034640 seconds
...

Installing the Software for an Enterprise Deployment

The procedure to install the software for an enterprise deployment is explained in this section.

- [Starting the Oracle SOA Suite Installer on SOAHOST1](#)
- [Navigating the Installation Screens](#)
- [Installing Oracle SOA Suite on the Other Host Computers](#)
- [Verifying the Installation](#)

Starting the Oracle SOA Suite Installer on SOAHOST1

To start the installation program:

1. Log in to SOAHOST1.
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 -jar Installer File Name
```

Be sure to replace the JDK location in these examples with the actual JDK location on your system.

Replace *Installer File Name* with the name of the actual installer file for your product listed in [Identifying and Obtaining Software Distributions for an Enterprise Deployment](#).

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

Navigating the Installation Screens

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

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

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you have already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. For more information about Oracle Fusion Middleware directory structure, see <i>Selecting Directories for Installation and Configuration in Planning an Installation of Oracle Fusion Middleware</i> .
Installation Type	Use this screen to select the type of installation and consequently, the products and feature sets you want to install. <ul style="list-style-type: none"> • Select SOA Suite

Screen	Description
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the documents in the Roadmap for Verifying Your System Environment section in <i>Installing and Configuring the Oracle Fusion Middleware Infrastructure</i> .
Installation Summary	Use this screen to verify the installation options that you selected. Click Install to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click Next when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click Finish to dismiss the installer.

Installing Oracle SOA Suite on the Other Host Computers

If you have configured a separate shared storage volume or partition for the products mount point and `ORACLE_HOME` on `SOAHOST2`, then you must also perform the product installation on `SOAHOST2`.

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 SOAHOST1](#) and [Navigating the Infrastructure Installation Screens](#) to create the Oracle home on the appropriate storage device.

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- [Reviewing the Installation Log Files](#)
- [Checking the Directory Structure](#)
- [Viewing the Contents of Your Oracle Home](#)

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see *Understanding Installation Log Files* in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

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

The addition of Oracle SOA Suite adds the following directory and sub-directories. Use the `ls --format=single-column` command to verify the directory structure.

```
ls --format=single-column $ORACLE_HOME/soa
```

```
bam
```

```
bin
bpm
common
integration
jlib
modules
plugins
readme.txt
reports
soa
```

For more information about the directory structure you should see after installation, see [What are the Key Oracle Fusion Middleware Directories?](#) in *Understanding Oracle Fusion Middleware*.

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See [Viewing the contents of an Oracle home](#) in *Installing Software with the Oracle Universal Installer*.

Creating the Oracle SOA Suite Database Schemas

Before you can configure an Oracle SOA Suite domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.

- [Starting the Repository Creation Utility \(RCU\)](#)
- [Navigating the RCU Screens to Create the Schemas](#)
- [Verifying Schema Access](#)
- [Configuring SOA Schemas for Transactional Recovery](#)

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Navigate to the `$ORACLE_HOME/oracle_common/bin` directory on your system.

```
cd $ORACLE_HOME/oracle_common/bin
```

2. Make sure that the `JAVA_HOME` environment variable is set to the location of a certified JDK on your system. The location should be up to but not including the `bin` directory. For example, if your JDK is located in `/u01/oracle/products/jdk`:

On UNIX operating systems:

```
export JAVA_HOME=/u01/oracle/products/jdk
```

3. Start RCU:

On UNIX operating systems:

```
./rcu
```

 **Note:**

If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces that are created by the RCU, provide the `-encryptTablespace true` option when you start RCU.

This defaults the appropriate RCU GUI Encrypt Tablespace checkbox selection on the Map Tablespaces screen without further effort during the RCU execution. See Encrypting Tablespaces in *Creating Schemas with the Repository Creation Utility*.

Navigating the RCU Screens to Create the Schemas

Schema creation involves the following tasks:

- [Task 1, Introducing RCU](#)
- [Task 2, Selecting a Method of Schema Creation](#)
- [Task 3, Providing Database Connection Details](#)
- [Task 4, Specifying a Custom Prefix and Selecting Schemas](#)
- [Task 5, Specifying Schema Passwords](#)
- [Task 6, Specifying Custom Variables](#)
- [Task 7, Verifying the Tablespaces for the Required Schemas](#)
- [Task 8, Creating Schemas](#)
- [Task 9, Reviewing Completion Summary and Completing RCU Execution](#)

Task 1 Introducing RCU

Click **Next**.

Task 2 Selecting a Method of Schema Creation

If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have the necessary privileges.

If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option generates a SQL script, which can be provided to your database administrator to create the required schema. See Understanding System Load and Product Load in *Creating Schemas with the Repository Creation Utility*.

Click **Next**.

Task 3 Providing Database Connection Details

Provide the database connection details for RCU to connect to your database.

1. In the **Database Type**, select **Oracle Database enabled for edition-based redefinition**.

 **Note:**

Oracle Database enabled for edition-based redefinition (EBR) is recommended to support Zero Down Time upgrades. For more information, see <https://www.oracle.com/database/technologies/high-availability/ebr.html>.

2. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
3. Enter the **Port** number of the RAC database scan listener, for example 1521.
4. Enter the RAC **Service Name** of the database.
5. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example SYS.
6. Enter the **Password** of the user name that you provided in step 4.
7. If you have selected the SYS user, ensure that you set the role to SYSDBA.
8. Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

Task 4 Specifying a Custom Prefix and Selecting Schemas

Choose **Select existing prefix**, and then select the prefix you used when you created the initial domain.

From the list of schemas, select the **SOA Suite** schema. This automatically selects **SOA Infrastructure**. In addition, the following dependent schemas have already been installed with the Infrastructure and are grayed out:

- **Common infrastructure Services**
- **Oracle Platform Security Services**
- **User Messaging Service**
- **Audit Services**
- **Audit Services Append**
- **Audit Services Viewer**
- **Metadata Services**
- **Weblogic Services**

The custom prefix is used to logically group these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

 **Tip:**

For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.

For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Click **Next** to proceed, then click **OK** on the dialog window to confirm that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords

Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database security requirements before you continue. RCU proceeds at this point even if you do not meet the password polices. Hence, perform this check outside RCU itself.

 **Tip:**

You must make a note of the passwords that you set on this screen; you need them later on during the domain creation process.

Click **Next**.

Task 6 Specifying Custom Variables

Specify the custom variables for the SOA Infrastructure schema.

For the enterprise deployment topology, enter `LARGE` for the **Database Profile** custom variable. See *About the Custom Variables Required for the SOA Suite Schemas in Installing and Configuring Oracle SOA Suite and Business Process Management*.

Click **Next**.

Task 7 Verifying the Tablespaces for the Required Schemas

On the Map Tablespaces screen, review the information, and then click **Next** to accept the default values.

Click **OK** in the confirmation dialog box.

Click **Next**.

Task 8 Creating Schemas

Review the summary of the schemas to be loaded, and click **Create** to complete schema creation.

 **Note:**

If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and recreate the schemas before you continue.

Task 9 Reviewing Completion Summary and Completing RCU Execution

When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

 **Note:**

If the database is a pluggable database (PDB), the appropriate tns alias that points to the PDB must be used in the sqlplus command.

```
./sqlplus FMW1412_SOAINFRA/<soainfra_password>

SQL*Plus: Release 23.0.0.0.0 - for Oracle Cloud and Engineered Systems on Wed Sep 11
14:20:00 2024 Version 23.5.0.24.07
Copyright (c) 1982, 2024, Oracle. All rights reserved.

Connected to:
Oracle Database 23ai EE Extreme Perf Release 23.0.0.0.0 - for Oracle Cloud and
Engineered Systems
Version 23.5.0.24.07

SQL>
```

Configuring SOA Schemas for Transactional Recovery

After you have installed the Oracle SOA Suite schemas successfully, use the procedure in this section to configure the schemas for transactional recovery.

This procedure sets the appropriate database privileges so that the Oracle WebLogic Server transaction manager can query the schemas for transaction state information and issue the appropriate commands, such as commit and rollback, during recovery of in-flight transactions after a WebLogic Server is unexpectedly unavailable.

These privileges should be granted to the owner of the SOAINFRA schema, which you defined when you created the schemas with the RCU.

To configure the SOA schemas for transactional recovery privileges:

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

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

2. If the database used by SOA is a pluggable database, change the session to connect to the PDB used. For example:

```
SQL> alter session set container=PDBNAME;
```

3. Enter the following commands:

```
SQL> Grant select on sys.dba_pending_transactions to soa_schema_prefix_soainfra;
```

```
Grant succeeded.
```

```
SQL> Grant force any transaction to soa_schema_prefix_soainfra;
```

```
Grant succeeded.
```

```
SQL>
```

Extending the Enterprise Deployment Domain with Oracle SOA Suite

Perform the following tasks to extend the existing enterprise deployment domain with the Oracle SOA Suite software.

Note:

For an improved footprint and to optimize startup, only core adapters are targeted to the SOA cluster (MFT Cluster if you are configuring MFT) after the Configuration Wizard session. You must target the second-tier adapters manually, if required. See [Targeting Adapters Manually](#).

Extending the domain involves the following tasks:

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens to Extend the Domain with Oracle SOA Suite](#)
Follow the instructions in these sections to extend the domain for Oracle SOA Suite.
- [Targeting Adapters Manually](#)
Only core adapters are targeted to the SOA cluster after you run the Configuration Wizard. You must target second-tier adapters manually, on a need basis.

Starting the Configuration Wizard

Note:

SSL store customizations were added to the `setUserOverridesLate.sh` in the domain creation chapter. Any customizations added to this file are preserved when a domain is extended and are carried over to remote servers when using the **pack** and **unpack** commands.

However, if you added any additional customizations to the `setDomainEnv.sh` script in the domain (such as custom libraries, JAVA command line options for starting the servers or environment variables), those will be overwritten by the configuration wizard when you extend the domain. Add all the startup parameters that apply to all servers in a domain to the `setUserOverridesLate.sh` file. This will preserve them across extensions.

To start the Configuration Wizard:

1. Stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.
2. Stop the Administration Server once all managed servers are in a steady state.

3. Run the following command to start the WebLogic Server Configuration Wizard.

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

Navigating the Configuration Wizard Screens to Extend the Domain with Oracle SOA Suite

Follow the instructions in these sections to extend the domain for Oracle SOA Suite.



Note:

This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- [Task 1, Selecting the Domain Type and Domain Home Location](#)
- [Task 2, Selecting the Configuration Template](#)
- [Task 3, Configuring High Availability Options](#)
- [Task 4, Specifying the Database Configuration Type](#)
- [Task 5, Specifying JDBC Component Schema Information](#)
- [Task 6, Providing the GridLink Oracle RAC Database Connection Details](#)
- [Task 7, Testing the JDBC Connections](#)
- [Task 8, Keystore](#)
- [Task 9, Selecting Advanced Configuration](#)
- [Task 10, Configuring Managed Servers](#)
- [Task 11, Configuring a Cluster](#)
- [Task 12, Assigning Server Templates](#)
- [Task 14, Assigning Managed Servers to the Cluster](#)
- [Task 15, Configuring Coherence Clusters](#)
- [Task 16, Verifying the Existing Machines](#)
- [Task 17, Assigning Servers to Machines](#)
- [Task 18, Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Task 19, Writing Down Your Domain Home and Administration Server URL](#)
- [Task 20, Start the Administration Server](#)

Task 1 Selecting the Domain Type and Domain Home Location

On the Configuration Type screen, select **Update an existing domain**.

In the **Domain Location** field, select the value of the `ASERVER_HOME` variable, which represents the complete path to the Administration Server domain home that you created in [Creating the Initial Infrastructure Domain for an Enterprise Deployment](#).

For more information about the directory location variables, see [File System and Directory Variables Used in This Guide](#).

For more information about the other options on this screen, see Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template

On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle SOA Suite Reference Configuration [soa]**

For more information about the options on this screen, see Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Note:

If you plan to extend the domain to add a component that does not support Reference Configuration, such as BPM and BAM (Reference Configuration is supported only in SOA, OSB, B2B, and ESS), the domain must be updated using the SOA classic template.

The classic SOA template, which does not implement the optimizations included in Reference Configuration is as follows:

- Oracle SOA Suite - 14.1.2.0.0. [soa]

Subsequent extensions on a Classic SOA domain for B2B or OSB must be done with Classic extension templates and not with Reference Configuration templates.

Task 3 Configuring High Availability Options

This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply HA options (that is, the Configuration Wizard creates the JDBC stores and configures ASM for them).

On the High Availability Options screen:

1. Select **Enable Automatic Service Migration** with **Database Basis**.
2. Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
3. Set **JMS Server Persistence** to **JMS JDBC Store**.

Note:

Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an Oracle database and makes resources available for all the servers in the cluster. So, the Configuration Wizard steps assume that the JDBC persistent stores are used along with Automatic Service Migration.

When you choose JDBC persistent stores, additional unused File Stores are automatically created but are not targeted to your clusters. Ignore these File Stores.

Click **Next**.

Task 4 Specifying the Database Configuration Type

On the Database Configuration Type screen, select **RCU Data**.

All fields are prepopulated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain. In the RCU Data screen:

- Verify that **Vendor** is Oracle and **Driver** is *Oracle's Driver (Thin) for Service Connections; Versions: Any.
- Verify that **Connection Parameters** is selected.
- Verify and ensure that credentials in all the fields are the same as those provided during the configuration of Oracle Fusion Middleware Infrastructure.

Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operation is successful.

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```



Tip:

For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*. For more information about the other options on this screen, see Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 5 Specifying JDBC Component Schema Information

On the JDBC Component Schema screen, select all the SOA schemas in the table. When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically. Click **Convert to GridLink**, and then click **Next**.

Task 6 Providing the GridLink Oracle RAC Database Connection Details

On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, <i>soaedg.example.com</i> .
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	These values are not required when you are using an Oracle 12c database or higher versions because the ONS list is automatically provided from the database to the driver.
Enable Fan	Verify that the Enable Fan check box is selected, so that the database can receive and process FAN events.

Task 7 Testing the JDBC Connections

Use the JDBC Component Schema Test screen to test the data source connections that 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.

For more information about the other options on this screen, see Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Keystore

Use this screen to specify details about the keystore to be used in the domain.

For a typical enterprise deployment, you can leave the default values.

See Keystore in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 9 Selecting Advanced Configuration

To complete domain configuration for the topology, select **Topology** on the Advanced Configuration screen.

Note:

JDBC stores are recommended and selected in [Task 3, Configuring High Availability Options](#) so there is no need to configure File Stores.

Task 10 Configuring Managed Servers

On the Managed Servers screen, a new Managed Server for Oracle SOA Suite appears in the list of servers. This server was created automatically by the Oracle SOA Suite configuration template that you selected in [Task 2, Selecting the Configuration Template](#).

Perform the following tasks to modify the default Oracle SOA Suite Managed Server and create a second Oracle SOA Suite Managed Server:

1. Rename the default Oracle SOA Suite Managed Server to `WLS_SOA1`.
2. Click **Add** to create a new Oracle SOA Suite Managed Server, and name it `WLS_SOA2`.

Tip:

The server names recommended here are used throughout this document; if you choose different names, be sure to replace them as needed.

3. Use the information in the following table to fill in the rest of the columns for each Oracle SOA Suite Managed Server.

For more information about the options on the Managed Server screen, see Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Administration Port	Server Groups
WLS_W SM1	SOAHOST1	Unchecked	Disabled	Checked	7010	9003	WSMPM-MAN-SVR and JRF-MAN-SVR
WLS_W SM2	SOAHOST2	Unchecked	Disabled	Checked	7010	9003	WSMPM-MAN-SVR and JRF-MAN-SVR
WLS_S OA1	SOAHOST1	Unchecked	Disabled	Checked	7004	9004	SOA-MGD-SVRS-ONLY
WLS_S OA2	SOAHOST2	Unchecked	Disabled	Checked	7004	9004	SOA-MGD-SVRS-ONLY

Task 11 Configuring a Cluster

In this task, you create a cluster of Managed Servers to which you can target the Oracle SOA Suite software.

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

For more information about the `soa.example.com` virtual server address, see [Configuring Virtual Hosts on the Hardware Load Balancer](#).

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `SOA_Cluster` in the **Cluster Name** field.
3. Specify `soa.example.com` in the **Frontend Host** field.
4. Leave the **Frontend HTTP Port** blank and use 443 (or your precise LBS listeners port for app requests) as the **Frontend HTTPS** port.

Note:

By default, server instances in a cluster communicate with one another by 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*.

For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 12 Assigning Server Templates

Click **Next** to continue.

Task 13 Configuring Dynamic Servers

Verify that all dynamic server options are **disabled** for clusters that are to remain as static clusters.

1. Ensure that **Calculated Machine Names** and **Calculated Listen Port** checkboxes on this screen are **unchecked**.
2. Ensure the **Server Template** and **Dynamic Server Groups** selections are **Unspecified**.
3. Click **Next**.

Task 14 Assigning Managed Servers to the Cluster

Use the Assign Servers to Clusters screen to assign WLS_SOA1 and WLS_SOA2 to the new cluster SOA_Cluster:

1. In the Clusters pane, select the cluster to which you want to assign the servers; in this case, SOA_Cluster.
2. In the Servers pane, assign WLS_SOA1 to SOA_Cluster by doing one of the following:
 - Click WLS_SOA1 Managed Server once to select it, and then click on the right arrow to move it beneath the selected cluster in the Clusters pane.
 - Double-click WLS_SOA1 to move it beneath the selected cluster in the clusters pane.
3. Repeat to assign WLS_SOA2 to SOA_Cluster.

For more information about the options on this screen, see Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 15 Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at 9991, as it was defined during the initial Infrastructure domain creation.

For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 16 Verifying the Existing Machines

Click **Next** to proceed.

Task 17 Assigning Servers to Machines

Use the Assign Servers to Machines screen to assign the Oracle SOA Suite Managed Servers you just created to the corresponding machines in the domain.

Assign WLS_SOA1 to SOAHOST1, and assign WLS_SOA2 to SOAHOST2.

For more information about the options on this screen, see Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 18 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains detailed configuration information for the domain that you are about to extend. 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 if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.

Click **Update** to execute the domain extension.

In the Configuration Progress screen, click **Next** when it finishes.

For more information about the options on this screen, see Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 19 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen shows the following items about the domain that you just configured, including:

- Domain Location
- Administration Server URL

Make a note of both these items, because you need them later; you need the domain location to access the scripts used to start the Administration Server, and you need the Administration Server URL to access the WebLogic Remote Console and Oracle Enterprise Manager Fusion Middleware Control.

Click **Finish** to dismiss the Configuration Wizard.

Task 20 Start the Administration Server

Start the Administration Server to ensure that the changes that you have made to the domain have been applied.

After you complete extending the domain with static clusters, go to [Targeting Adapters Manually](#).

Targeting Adapters Manually

Only core adapters are targeted to the SOA cluster after you run the Configuration Wizard. You must target second-tier adapters manually, on a need basis.

The following second-tier adapters have to be targeted manually:



Note:

Some of these adapters may not be available with the default installation. See [Oracle Technology Network for Adapter availability](#).

- MSMQAdapter
- SocketAdapter
- OracleBamAdapter
- CoherenceAdapter
- SAPAdapter
- SiebelAdapter
- ERPAdapter
- Oracle SalesCloudAdapter
- RightNowAdapter
- EloquaAdapter
- NetSuiteAdapter
- LdapAdapter
- JDEWorldAdapter

- CloudSDK

To target a second-tier adapter manually:

1. Navigate to **Edit Tree > Deployments > App Deployments**.
2. Locate and click the name of the adapter in the Deployments table.
3. In the **Targets** tab, select **SOA_Cluster** and move it to the "Chosen" pane.

 **Note:**

If you are deploying MFT, select MFT_Cluster as the target.

4. Click **Save**.
The cart on the top right part of the screen will show **full** with a yellow bag inside.
5. Click the Cart icon on the top right and select **Commit Changes**.
6. In the navigation tree pane of the console, navigate to **Monitoring Tree > Deployments > Ap Development Runtimes** and verify that the adapter is in the Active state.

Update Certificates for New Frontend Addresses

This section contains information about certificates for new frontend addresses.

 **Note:**

About Certificates for the domain extension.
Since the SOA and WSM servers use the same listen addresses (different port), there is no need to create new certificates and update stores. However, the SOA cluster uses a different front end address that will be added as a trusted endpoint in the [Configuring the Web Tier for the Extended Domain](#).

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in the [Updating the WebLogic Servers Security Settings](#) and update SSL settings for the WLS_SOA1 and WLS_SOA2 servers.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the Oracle SOA Suite instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and machines.

[Table 12-2](#) summarizes the steps required to propagate the changes to all the domain directories and machines.

Note that there is no need to propagate the updated domain to the WEBHOST1 and WEBHOST2 machines because there are no changes to the Oracle HTTP Server instances on those host computers.

Table 12-2 Summary of Tasks Required to Propagate the Domain Changes to Domain Directories and Machines

Task	Description	More Information
Pack up the Extended Domain on SOAHOST1	Use the <code>pack</code> command to create a new template JAR file that contains the new Oracle SOA Suite Managed Servers configuration. When you pack up the domain, create a template JAR file called <code>soadomaintemplate.jar</code> .	Packing Up the Extended Domain on SOAHOST1
Unpack the Domain in the Managed Servers directory on SOAHOST1	Unpack the template JAR file in the Managed Servers directory on SOAHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1
Unpack the Domain on SOAHOST2	Unpack the template JAR file in the Managed Servers directory on the SOAHOST2 local storage.	Unpacking the Domain on SOAHOST2

- [Packing Up the Extended Domain on SOAHOST1](#)
- [Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1](#)
- [Unpacking the Domain on SOAHOST2](#)

Packing Up the Extended Domain on SOAHOST1

Use the following steps to create a template JAR file that contains the domain configuration information:

1. Log in to SOAHOST1 and run the `pack` command to create a template JAR file as follows:

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

./pack.sh -managed=true \
  -domain=ASERVER_HOME \
  -template=full_path/soadomaintemplate.jar \
  -template_name=soa_domain_template \
  -log=/tmp/pack_soa.log \
  -log_priority=debug
```

In this example:

- Replace `ASERVER_HOME` with the actual path to the domain directory that you created on the shared storage device.
- Replace `full_path` with the complete path to the directory where you want the template jar file saved.
- `soadomaintemplate.jar` is a sample name for the JAR file that you are creating, which contains the domain configuration files, including the configuration files for the Oracle HTTP Server instances.
- `soa_domain_template` is the name assigned to the domain template file.

2. Make a note of the location of the template 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*](#).

Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1

To copy the updated domain configuration information from the Administration Server domain directory to the Managed Servers domain directory:

1. Log in to SOAHOST1 if you haven't already.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the SOAHOST1 local storage device.

Use the examples in [File System and Directory Variables Used in This Guide](#) as a guide.

3. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

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

./unpack.sh -domain=MSERVER_HOME \
  -overwrite_domain=true \
  -template=/full_path/soadomaintemplate.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/soadomaintemplate.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device.
- Replace `APPLICATION_HOME` with the complete path to the applications 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*](#).

4. Change directory to the newly created `MSERVER_HOME` directory and verify that the domain configuration files were copied to the correct location on the SOAHOST1 local storage device.

Unpacking the Domain on SOAHOST2

This procedure assumes you have copied the file that you created earlier in a location that is accessible from both SOAHOST1 and SOAHOST2; such as the `ASERVER_HOME` directory, which is located on the shared storage filer:

1. Log in to SOAHOST2.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the SOAHOST2 storage device.

Use the examples in [File System and Directory Variables Used in This Guide](#) as a guide.

3. Make sure the `create_domain.jar` accessible to SOAHOST2.

For example, if you are using a separate shared storage volume or partition for SOAHOST2, then copy the template to the volume or partition mounted to SOAHOST2.

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/create_domain.jar \
            -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME
```

 **Note:**

The `-overwrite_domain` option in the `unpack` command allows unpacking a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this `unpack` operation.

Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional java command line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when using the `pack` and `unpack` commands.

In this example:

- Replace `MSERVER_HOME` with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain will be unpacked.
- Replace `/full_path/create_domain.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device.
- Replace `APPLICATION_HOME` with the complete path to the Application directory for the domain on shared storage. See [File System and Directory Variables Used in This Guide](#).

 **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 `SOAHOST2` local storage device.

Starting and Validating the WLS_SOA1 Managed Server

Now that you have extended the domain, started the Administration Server, and propagated the domain to the other hosts, you can start the newly configured Oracle SOA Suite Managed Servers.

This process involves three tasks as described in the following sections.

- [Starting the WLS_SOA1 Managed Server](#)
- [Adding the SOAAdmin Role to the Administrators Group](#)
- [Validating the Managed Server by Logging in to the SOA Infrastructure](#)

Starting the WLS_SOA1 Managed Server

Note:

SOA Servers depend on the policy access service to be functional. This implies that the WSM-PM Managed Servers in the domain need to be up and running and reachable before the SOA servers are started.

To start the WLS_SOA1 Managed Server:

1. Enter the following URL into a browser to display the Fusion Middleware Control login screen:
`https://admin.example.com:445/em`
2. Sign in to the Fusion Middleware Control by using the administrator's account. For example: `weblogic_soa`.
3. In the **Target Navigation** pane, expand the domain to view the Managed Servers in the domain.
4. Select only the **WLS_SOA1** Managed Server and click **Start Up** on the Oracle WebLogic Server toolbar.
5. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_SOA1 Managed Server is up and running.

Adding the SOAAdmin Role to the Administrators Group

Before you validate the Oracle SOA Suite configuration on the WLS_SOA1 Managed Server, add the `SOAAdmin` administration role to the enterprise deployment administration group (`SOA Administrators`).

To perform this task, refer to [Configuring Roles for Administration of an Enterprise Deployment](#).

Validating the Managed Server by Logging in to the SOA Infrastructure

After you add the `SOAAdmin` role to the `SOA Administrators` group, you can then validate the configuration of the Oracle SOA Suite software on the WLS_SOA1 Managed Server as follows:

1. Use your web browser to navigate to the following URL (notice the slash at the end of the url to avoid any redirections):
`https://SOAHOST1:7004/soa-infra/`
2. Log in by using the enterprise deployment administrator user credentials (`weblogic_soa`).
You should see a web page with the following title:

Welcome to the Oracle SOA Platform on WebLogic

Starting and Validating the WLS_SOA2 Managed Server

After you validate the successful configuration and startup of the WLS_SOA1 Managed Server, you can start and validate the WLS_SOA2 Managed Server.

To start and validate the WLS_SOA2 Managed Server, use the procedure in [Starting and Validating the WLS_SOA1 Managed Server](#) for WLS_SOA2 Managed Server.

For validation of the URL, enter the following URL in your web browser and log in by using the enterprise deployment administrator user (`weblogic_soa`):

`https://SOAHOST2:7004/soa-infra/`

Modifying the Upload and Stage Directories to an Absolute Path

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. See [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).

Configuring the Web Tier for the Extended Domain

Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.

- [Generate the Required Certificates for OHS SSL Listeners](#)
- [Configuring Oracle HTTP Server for the WLS_SOA Managed Servers](#)
- [Validating the Oracle SOA Suite URLs Through the Load Balancer](#)

Generate the Required Certificates for OHS SSL Listeners

Follow the steps described in the [Generate Required Certificates for OHS SSL Listeners](#) section in [Starting the Oracle HTTP Server Instances](#) to add the new fronted address to the certificate stores and update the SAN for the OHS listeners certs.

When asked to replace the existing OHS Virtual Host certificates, answer yes so that they are updated with the new frontend address for the SOA cluster as SAN.

Configuring Oracle HTTP Server for the WLS_SOA Managed Servers

To configure the Oracle HTTP Server instances in the web tier so that they route requests correctly to the Oracle SOA Suite cluster, use the following procedure to create an additional Oracle HTTP Server configuration file that creates and defines the parameters of the `soa.example.com` virtual server.

This procedure assumes that you performed the Oracle HTTP Server configuration tasks described in [Configuring Oracle HTTP Server to Route Requests to the Application Tier](#).

Create a `soa_vh.conf` file by copying the existing `admin_vh.conf` file. This will transfer most of the required SSL configuration. Then update it with the entries required by SOA:

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

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

2. Copy the existing admin_vh.conf file to the soa_vh.conf file. This will transfer most of the required SSL configuration:

```
cp admin_vh.conf soa_vh.conf
```

3. Edit the file and customize with the required values for **Listen**, **ServerName**, **VirtualHost**, **SSLWallet** and **Location** directives (AllowEncodedSlashes not needed here).

 **Note:**

For the Listen address you need to specify a different port from the ones used in previous virtual hosts (admin_vh.conf and soainternal_vh.conf). Otherwise the listeners will conflict.

```
#[Listen] OHS_SSL_PORT

Listen WEBHOST1:4443
##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost WEBHOST1:4443>
ServerName soa.example.com:443
  <IfModule ossl_module>

    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # Client Authentication (Type):
    # Client certificate verification type and depth. Types are
    # none, optional and require.
    SSLVerifyClient None

    # SSL Protocol Support:
    # Configure usable SSL/TLS protocol versions.
    SSLProtocol TLSv1.2 TLSv1.3

    # Option to prefer the server's cipher preference order
    SSLHonorCipherOrder on

    # SSL Cipher Suite:
    # List the ciphers that the client is permitted to negotiate.
    SSLCipherSuite
    TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,TLS_ECDHE_
    ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_RSA_W
    ITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

    #Path to the wallet
    #SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/${COMPONENT_TYPE}/
    instances/${COMPONENT_NAME}/keystores/default"
    SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1"
```

```
<FilesMatch "\.(cgi|shtml|phtml|php)$">
  SSLOptions +StdEnvVars
</FilesMatch>

<Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/${COMPONENT_TYPE}/
instances/${COMPONENT_NAME}/cgi-bin">
  SSLOptions +StdEnvVars
</Directory>
  BrowserMatch "MSIE [2-5]" \
  nokeepalive ssl-unclean-shutdown \
  downgrade-1.0 force-response-1.0

# Add the following directive to add HSTS
<IfModule mod_headers.c>
  Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
</IfModule>

<Location /soa-infra>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# SOA inspection.wsil
<Location /inspection.wsil>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# Worklist
<Location /integration>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# UMS prefs
<Location /sdpmessaging/userprefs-ui>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# Default to-do taskflow
<Location /DefaultToDoTaskFlow>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# Workflow
<Location /workflow>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

#Required if attachments are added for workflow tasks
<Location /ADFAttachmentHelper>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# SOA composer application
```

```

<Location /soa/composer>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

</IfModule>
</VirtualHost>

```

 **Note:**

- The URL entry for `/workflow` is optional. It is for the workflow tasks that are associated with Oracle ADF task forms. The `/workflow` URL itself can be a different value, depending on the form.
- The `WebLogicCluster` directive needs only a sufficient number of redundant `server:port` combinations to guarantee an initial contact in case of a partial outage. The actual total list of cluster members is retrieved automatically on the first contact with any given node.

4. Copy the `soa_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (`ohs2`):

```
$WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf/
```

5. Edit the `soa_vh.conf` file and change any references to `WEBHOST1` to `WEBHOST2` in the `<VirtualHost>` directives.
6. Restart the Oracle HTTP servers on `WEBHOST1` and `WEBHOST2`.

 **Note:**

If internal invocations are going to be used in the system, add the appropriate locations to the `soainternal` virtual host.

Validating the Oracle SOA Suite URLs Through the Load Balancer

To validate the configuration of the Oracle HTTP Server virtual hosts and to verify that the hardware load balancer can route requests through the Oracle HTTP Server instances to the application tier:

1. Verify that the server status is reported as **Running** in the WebLogic Remote Console or in the Fusion Middleware Control.

If the server is shown as **Starting** or **Resuming**, wait for the server status to change to **Started**. If another status is reported (such as **Admin** or **Failed**), check the server output log files for errors.

2. Verify that you can access these URLs:
 - `https://soa.example.com:443/soa-infra`
 - `https://soa.example.com:443/integration/worklistapp`
 - `https://soa.example.com:443/sdpMessaging/userprefs-ui`

- <https://soa.example.com:443/soa/composer>
3. Verify that Identity Service can be invoked successfully on the application tier by accessing the following load balancer URL:

<https://soa.example.com:443/integration/services/IdentityService/identity?WSDL>

Post-Configuration Steps for Oracle SOA Suite

After you install and configure Oracle SOA Suite, consider the following post-configuration tasks.

- [Configuring Oracle Adapters for Oracle SOA Suite](#)
- [Considerations for Sync-Async Interactions in a SOA Cluster](#)
- [Updating FusionAppsFrontendHostUrl](#)

Configuring Oracle Adapters for Oracle SOA Suite

If the Oracle SOA Suite applications that you are developing take advantage of any of the Oracle adapters for Oracle SOA Suite, then you should make sure that the adapters are configured to work efficiently and securely in the enterprise topology.

See the following topics for more information.

- [Enabling High Availability for Oracle File and FTP Adapters](#)
- [Enabling High Availability for Oracle JMS Adapters](#)
- [Enabling High Availability for the Oracle Database Adapter](#)

Enabling High Availability for Oracle File and FTP Adapters

If the Oracle SOA Suite applications that you are developing or deploying require the Oracle File and FTP Adapters, you must configure the adapters for high availability in the enterprise deployment topology.

Use the following sections to complete this task.

- [Understanding the Oracle File and FTP Adapter Configuration](#)
- [Configuring the Oracle File Adapter in the Remote Console](#)
- [Editing the JCA File Within the Composite Application](#)
- [Configuring the Oracle FTP Adapter](#)

Understanding the Oracle File and FTP Adapter Configuration

The Oracle File and FTP adapters enable a BPEL process or an Oracle Mediator to read and write files on private file systems and on remote file systems through the File Transfer Protocol (FTP).

When configured properly, these adapters support high availability for an active-active topology with Oracle BPEL Process Manager and Oracle Mediator service engines for both inbound and outbound operations.

For general information about this task, see *Configuring Oracle File and FTP Adapters in Understanding Technology Adapters*. The instructions provided here are specific to the Oracle SOA Suite enterprise deployment.



Note:

The File Adapter picks up a file from the inbound directory, processes it, and then outputs a file to the output directory. Because the File Adapter is non-transactional, files can be processed twice. As a result, it is possible to get duplicate files when there is failover in the RAC backend or in the SOA managed servers.

Configuring the Oracle File Adapter in the Remote Console

To make the Oracle File Adapter highly available, first modify the Oracle File Adapter deployment descriptor for the connection-instance that corresponds to `eis/HFileAdapter`.

To configure adapters, perform the following steps in the WebLogic Remote Console:

1. Create a deployment plan directory on shared storage (if it does not exist) as follows:

```
mkdir -p $DEPLOY_PLAN_HOME/soaedg_domain
```

2. Create a fileadapter control directory in the shared runtime folder as follows:

```
mkdir -p /u01/oracle/runtime/soaedg_domain/SOA_Cluster/fadapter
```

3. In the **Monitoring Tree**, navigate to **Deployments > Application Management > File Adapter**.
4. Click **Create Plan** (if it does not already have a plan) and use the `DEPLOY_PLAN_HOME/domain_name/` as its directory.
5. After the new plan is displayed under the **File Adapter**, in the **Monitoring Tree** navigate to **Deployments > Application Management > File Adapter**.
6. Select **Configuration > Outbound Connection Pool Groups**.
7. Navigate to **javax.resource.cci.ConnectionFactory > Outbound Connection Pool Instances**.
8. Navigate to **eis/HFileAdapter > Properties**.
9. Modify the values of the properties described in the following table:

Table 12-3 The following table describes modified parameters

Parameter	Description
controlDir	Enter the directory where you want the control files to be stored. You must set it to a shared location if multiple WebLogic Server instances run in a cluster. Structure the directory for shared storage as follows: <code>ORACLE_RUNTIME/domain_name/cluster_name/fadapter</code>
inboundDataSource	Set the value to <code>jdbc/SOADDataSource</code> .
outboundDataSource	Set the value to <code>jdbc/SOADDataSource</code> .

Table 12-3 (Cont.) The following table describes modified parameters

Parameter	Description
outboundDataSourceLocal	Set the value to <code>jdbc/SOALocalTxDataSource</code> . This is the data source where the schemas that corresponds to high availability are precreated.
outboundLockTypeForWrite	Set the value to <code>oracle</code> if you are using Oracle Database. By default the Oracle File and FTP Adapters use an in-memory mutex to lock outbound write operations. You must choose from the following values for synchronizing write operations: <ul style="list-style-type: none"> <code>memory</code>: The Oracle File and FTP Adapters use an in-memory mutex to synchronize access to the file system. <code>oracle</code>: The adapter uses Oracle Database sequence. <code>db</code>: The adapter uses a pre-created database table (<code>FILEADAPTER_MUTEX</code>) as the locking mechanism. You must use this option only if you are using a schema other than the Oracle Database schema. <code>user-defined</code>: The adapter uses a user-defined mutex. To configure the user-defined mutex, you must implement the mutex interface: <code>oracle.tip.adapter.file.Mutex</code> and then configure a new binding-property with the name <code>oracle.tip.adapter.file.mutex</code> and value as the fully qualified class name for the mutex for the outbound reference.
workingDirectory	Retain the default value.

10. Redeploy the Adapter using the console.
 - a. In the **Monitoring Tree**, navigate to **Deployments > Application Management**.
 - b. Select the **FileAdapter deployment** check box.
 - c. Click **Update/Redeploy > Redeploy - Deployment Source and Plan on Server** (it is not possible to use **Update - Deployment Plan on Server** because these are non-dynamic changes).

Ensure that the deployment plan is correct in the Plan Path filed.

11. Click **Done**.
Wait for the operation to complete.
12. After the operation is complete, check the values entered in the **Monitoring > Deployments > Application Management > FileAdapter > Deployment plan**.

Editing the JCA File Within the Composite Application

After you have configured the FileAdapter deployment in the Remote Console, you can edit the `.jca` file that is included in the composite applications to be deployed so that they can use the connection factory that was configured in the previous steps, as shown in [Example 12-1](#).



Note:

The location attribute is set to `eis/HAFileAdapter` for the connection factory.

Example 12-1 Example of the File Adapter .JCA File Modifications for an Enterprise Deployment

```
<adapter-config name="FlatStructureOut"
  adapter="File Adapter"
  xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
  <connection-factory location="eis/HAFileAdapter" adapterRef=""/>
  <endpoint-interaction portType="Write_ptt"
```



```
        operation="Write">
<interaction-spec className="oracle.tip.adapter.file.outbound.FileInteractionSpec">
    <property../>
    <property../>
    </interaction-spec>
</endpoint-interaction>
</adapter-config>
```

Configuring the Oracle FTP Adapter

If your application requires an FTP Adapter, then repeat the procedures [Configuring the Oracle File Adapter in the Remote Console](#) and [Editing the JCA File Within the Composite Application](#), with the following differences:

- Locate the **FtpAdapter** deployment in the list of deployments in the Remote Console.
- Click **FtpAdapter** to display the Settings for the FtpAdapter page.
- Use **eis/Ftp/HAFtpAdapter** as connection factory.
- Use as **ControlDir** the following location:

```
ORACLE_RUNTIME/domain_name/cluster_name/ftpadapter
```

- Enter a shared storage location for the deployment plan. The directory structure is as follows:

```
DEPLOY_PLAN_HOME/soaedg_domain/FtpAdapterPlan.xml
```

- Update the FTPAdapter deployment in the console. See [Configuring the Oracle File Adapter in the Remote Console](#).

Enabling High Availability for Oracle JMS Adapters

When the Oracle JMS adapter communicates with multiple servers in a cluster, the adapter's connection factory property `FactoryProperties` must list available servers. If it does not list servers, the connection is established to only one random server. If that particular server goes down, no further messages are processed.

To avoid this issue, you can use the “cluster name” syntax in the `FactoryProperties` of the adapter instead of using the static list of members. The cluster name syntax is as follows:

```
cluster:t3s://cluster_name
```

When you use `cluster:t3s://cluster_name`, the invocation fetches the complete list of members in the cluster at any given time, thus avoiding any dependencies on the initial servers and accounting for every member that is alive in the cluster at that point of time. Note that you can use this cluster syntax only when the cluster is in the same domain.

1. Create a deployment plan directory on shared storage (if it does not exist) as follows:

```
mkdir -p $DEPLOY_PLAN_HOME/soaedg_domain
```

2. In the **Monitoring Tree**, navigate to **Deployments > Application Management > JMS Adapter**.
3. Create Plan (if it does not already have a plan) and use the `DEPLOY_PLAN_HOME/domain_name/` as its directory.
4. After the new plan is displayed under the **JMS Adapter**, in the **Monitoring Tree** navigate to **Deployments > Application Management > JMS Adapter**.

5. Navigate to **Configuration > Outbound Connection Pool Groups**.
6. Navigate to **oracle.tip.adapter.jms.IJmsConnectionFactory> Outbound Connection Pool Instances**.
7. Click **eis/wls/Queue > Properties**.
8. Click the **FactoryProperties** field (click the corresponding cell under Property value), enter the following, all in one line, separated by semicolons. Adjust the values to match your cluster name, username and password:

```
java.naming.factory.initial=weblogic.jndi.WLInitialContextFactory;
java.naming.provider.url=cluster:t3s://SOA_Cluster;
java.naming.security.principal=soaedgadmin;
java.naming.security.credentials=<password>
```

9. Click **Save** after you update the properties.
10. Redeploy the Adapter using the console.
 - a. Navigate to **Monitoring > Deployments > Application Management**.
 - b. Select the **JMSAdapter deployment** check box.
 - c. Click **Update/Redeploy > Redeploy - Deployment Source and Plan on Server** (not possible to use **Update - Deployment Plan on Server** because these are non dynamic changes)

Ensure that the deployment plan is correct in the Plan Path filed.

11. Click **Done**.
Wait for the operation to complete.
12. After the operation is complete, check the values entered in the **Monitoring > Deployments > Application Management > JMSAdapter > Deployment plan**.

Enabling High Availability for the Oracle Database Adapter

To ensure High Availability while leveraging the Oracle Database Adapter, the Logical Delete Polling Strategy is used normally as it performs better than a physical delete. However, when you have a clustered environment where multiple nodes are polling for the same data, a single record might get processed more than once. To avoid this problem, Oracle Database Adapter uses a distributed polling technique that uses an Oracle Database feature called skip locking.

If you were using the Logical Delete Polling Strategy approach previously, you can remove (in `db.jca`) or clear (Logical Delete Page of wizard) the `MarkReservedValue`, and you automatically get skip locking.

The benefits of using skip locking over a reserved value include:

- Skip locking scales better in a cluster and under load.
- All work is in one transaction (as opposed to update/reserve, then commit, then select in a new transaction), so the risk of facing a non-recoverable situation in a high availability environment is minimized.
- No unique `MarkReservedValue` must be specified. Previously, for this to work you would have to configure a complex variable, such as `R${weblogic.Name-2}-${IP-2}-${instance}`.

If you are using Logical Delete polling, and you set `MarkReservedValue`, skip locking is not used.

For more information, see "Scalability" and "Polling Strategies" in the Oracle Fusion Middleware User's Guide for Technology Adapters.

Considerations for Sync-Async Interactions in a SOA Cluster

In a SOA cluster, the following scenarios are not supported:

- Synchronous BPEL process with mid-process receive.
- Synchronous BPEL process calling asynchronous services.
- Callback from synchronous processes.

Updating FusionAppsFrontendHostUrl

You must configure Oracle Workflow with the appropriate URL so that the default-to-do tasks and custom tasks' details use the front-end load balancer to create task-display URLs. To configure the appropriate URLs:

1. Log into Oracle Enterprise Manager Fusion Middleware Control with the administrator username and password.
2. In the left navigation tree, expand **WebLogic Domain**, and then click **System MBean Browser**.
3. Navigate to **Application Defined Mbean > oracle.as.soainfra.config > WLS_SOA1 > WorkflowConfig > human-workflow**.

Note:

In a clustered environment, there are multiple human-workflow Mbeans, one for every server in the cluster. Modify any one of them to update the property centrally in MDS for the entire cluster.

4. On the right panel, look for the **FusionAppsFrontendHostUrl** attribute.
5. For the **FusionAppsFrontendHostUrl** attribute, specify the value `*=https://soa.example.com:443`.
6. Click **Apply**.

Replacing Connect Strings with the Appropriate TNS Alias

Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

For more information about how to use TNS alias in your Datasources, see [Using TNS Alias in Connect Strings](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the

installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see [Performing Backups and Recoveries for an Enterprise Deployment](#).

Extending the Domain with Oracle Service Bus

The procedures described in this chapter guide you through the process of extending the enterprise deployment topology with Oracle Service Bus (OSB).

- [About Configuring Oracle Service Bus in Its Own Domain](#)
When you add Oracle Service Bus (OSB) to your enterprise topology, you can add it to the existing SOA domain, or you can create a new domain for OSB, separate from the Oracle SOA Suite domain.
- [Variables Used When Configuring Oracle Service Bus](#)
As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.
- [Overview of Adding OSB to the Topology](#)
Before you add OSB to the topology, you must ensure that you have already performed the steps that are required to create an initial Infrastructure domain and then extended the domain to include Oracle SOA suite.
- [Prerequisites for Extending the Domain to Include Oracle Service Bus](#)
Before you extend the current domain, ensure that your existing deployment meets the necessary prerequisites.
- [Installing Oracle Service Bus Software](#)
You can install Oracle Service Bus in an enterprise deployment by using the OSB Installer.
- [Extending the SOA or Infrastructure Domain to Include Oracle Service Bus](#)
You can use the Configuration Wizard to extend the existing enterprise deployment SOA domain with the Oracle Service Bus. You have to perform a series of additional tasks to complete the extension.
- [Update Certificates for New Frontend Addresses](#)
This section contains information about certificates for new frontend addresses.
- [Update the WebLogic Servers Security Settings](#)
This section contains information about WebLogic Servers security settings.
- [Propagating the Extended Domain to the Domain Directories and Machines](#)
After you have extended the domain with the OSB instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and systems.
- [Modifying the Upload and Stage Directories to an Absolute Path](#)
- [Configuring the Web Tier for the Extended Domain](#)
It is important to understand how to configure the web server instances on the web tier so that they route requests for both public and internal URLs to the proper clusters in the extended domain.
- [Post-Configuration Tasks for Oracle Service Bus](#)
After you install and configure Oracle Service Bus in the domain, consider the following post-configuration tasks.
- [Replacing Connect Strings with the Appropriate TNS Alias](#)
Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

- [Backing Up the Configuration](#)
It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

About Configuring Oracle Service Bus in Its Own Domain

When you add Oracle Service Bus (OSB) to your enterprise topology, you can add it to the existing SOA domain, or you can create a new domain for OSB, separate from the Oracle SOA Suite domain.

For more information about the OSB topology, see [About the Topology Options for Oracle Service Bus](#).

If you decide to configure Oracle Service Bus in a separate domain, then keep in mind the following when you use the instructions to add Oracle Service Bus to your topology:

- Ignore any references to the SOA Managed Servers or the SOA Cluster. These elements of the domain only exist if you extend a domain that has already been extended with Oracle SOA Suite.
- You must run the RCU to create the SOAINFRA schema for the Oracle Service Bus domain. This schema is required by Oracle Service Bus. You must use a unique SOAINFRA schema and schema prefix for the Oracle Service Bus domain.
- When you run the Configuration Wizard, the High Availability Options screen appears as described in [Navigating the Configuration Wizard Screens to Extend the Domain with Oracle SOA Suite](#).

This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply HA options (that is, the Configuration Wizard creates the JDBC stores and configures ASM for them).

Oracle recommends that you select the following options:

- Select **Enable Automatic Service Migration with Database Leasing**.
- Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
- Set **JMS Server Persistence** to **JMS JDBC Store**.

Variables Used When Configuring Oracle Service Bus

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

The values for several directory variables are defined in [File System and Directory Variables Used in This Guide](#).

- `ORACLE_HOME`
- `ASERVER_HOME`
- `MSERVER_HOME`
- `JAVA_HOME`
- `WEB_DOMAIN_HOME`

In addition, you reference the following virtual hostname addresses that are defined in [Physical and Virtual IP Addresses Required by the Enterprise Topology:ADMINVHN](#)

Actions in these topics are performed on the following host computers:

- SOAHOST1
- SOAHOST2
- WEBHOST1
- WEBHOST2

Overview of Adding OSB to the Topology

Before you add OSB to the topology, you must ensure that you have already performed the steps that are required to create an initial Infrastructure domain and then extended the domain to include Oracle SOA suite.

[Table 13-1](#) lists and describes the high-level steps to extend an existing SOA domain or an existing Infrastructure domain for Oracle Service Bus.

Table 13-1 Steps for Extending a SOA Domain to Include Oracle Service Bus

Step	Description	More Information
Install Oracle Service Bus software.	Install OSB software on the target system.	Installing Oracle Service Bus Software
Optionally, install the SOAINFRA schema in a supported database.	OSB requires the SOAINFRA schema for the <code>wlsbjmsrpDataSource</code> data source. If you plan to run OSB in its own domain, then you must be sure that you have installed a separate SOAINFRA schema for OSB in a supported database. Be sure to use a unique schema for the SOAINFRA schema that is used by the OSB domain.	Creating the Oracle SOA Suite Database Schemas
Optionally, create a new Infrastructure domain.	If you plan to run OSB in its own domain, then you must first create an Infrastructure domain, so you can extend that domain with OSB.	Creating the Initial Infrastructure Domain for an Enterprise Deployment
Run the Configuration Wizard to Extend the Domain.	Extend the SOA or Infrastructure domain to contain Oracle Service Bus components.	Extending the SOA or Infrastructure Domain to Include Oracle Service Bus
Propagate the Domain Configuration to the Managed Server Directory in SOAHOST1 and to SOAHOST2.	Oracle Service Bus requires some updates to the WebLogic Server start scripts. Propagate these changes by using the <code>pack</code> and <code>unpack</code> commands.	Propagating the Extended Domain to the Domain Directories and Machines
Start the Oracle Service Bus Servers.	Oracle Service Bus servers extend an already existing domain. As a result, the Administration Server and respective Node Managers are already running in SOAHOST1 and SOAHOST2.	Starting and Validating the WLS_OSB1 Managed Server

Table 13-1 (Cont.) Steps for Extending a SOA Domain to Include Oracle Service Bus

Step	Description	More Information
Validate the WLS_OSB Managed Servers.	Verify that the server status is reported as Running in the Admin Console and access URLs to verify status of servers.	Starting and Validating the WLS_OSB2 Managed Server
Configuring Oracle HTTP Server for the WLS_OSBn Managed Servers.	To enable Oracle HTTP Server to route to Oracle Service Bus console and Oracle Service Bus service, set the WebLogicCluster parameter to the list of nodes in the cluster.	Configuring Oracle HTTP Server for the Oracle Service Bus
Validating Access Through Oracle HTTP Server.	Verify that the server status is reported as Running.	Validating the Oracle Service Bus URLs Through the Load Balancer
Enable High Availability for Oracle File and FTP Adapters.	Make Oracle File and FTP Adapters highly available for outbound operations by using the database mutex locking operation.	Enabling High Availability for Oracle DB_ File and FTP Adapters
Backing up the Oracle Service Bus Configuration.	To back up the domain configuration for immediate restoration in case of failures in future procedures.	Backing Up the Configuration

Prerequisites for Extending the Domain to Include Oracle Service Bus

Before you extend the current domain, ensure that your existing deployment meets the necessary prerequisites.

- Back up the installation. If you have not yet backed up the existing Fusion Middleware Home and domain, Oracle recommends backing it up now.
To back up the existing Fusion Middleware Home and domain, see [Performing Backups and Recoveries in the SOA Enterprise Deployments](#).
- Verify that you have installed the Infrastructure and SOA software binaries in an Oracle home on shared storage and they are available from SOAHOST1 and SOAHOST2.
- If Oracle Service Bus is being configured in the same domain as SOA, then the appropriate SOAINFRA schema (used by the wlsbjmsrpDataSource) is already available. If OSB is being configured in its own domain, then you must run RCU to install the SOAINFRA schema in a supported database by using a different schema prefix than the SOAINFRA schema used by the SOA domain.
- You have already configured Node Manager, Administration Server, (optionally SOA Servers) and WSM Servers as described in previous chapters to run a SOA system. Optionally, you may have already configured Server migration, transaction logs, coherence, and all other configuration steps for the SOA System.
- If you haven't done so already, verify that the system clocks on each host computer are synchronized. You can do this by running the date command simultaneously on the hosts in each cluster.

Alternatively, there are third-party and open-source utilities you can use for this purpose.

Installing Oracle Service Bus Software

You can install Oracle Service Bus in an enterprise deployment by using the OSB Installer.

- [Starting the Oracle Service Bus Installer](#)
- [Navigating the OSB Installation Screens](#)
- [Installing the Software on Other Host Computers](#)
- [Validating the OSB Installation](#)

Starting the Oracle Service Bus Installer

To start the installation program, perform the following steps.

1. Log in to the target system, SOAHOST1.
2. Go to the directory in which you downloaded the installation program.
3. Set the `path` for the java executable:

```
export JAVA_HOME=JAVA_HOME
export PATH=$JAVA_HOME/bin:$PATH
```

In this example, replace `JAVA_HOME` with the value this variable listed in [File System and Directory Variables Used in This Guide](#) and entered in the *Enterprise Deployment Workbook*.

4. Launch the installation program by entering the following command:

```
java -jar fmw_14.1.2.0.0_osb.jar
```

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

Navigating the OSB Installation Screens

[Table 13-2](#) provides description of each installation program screen.

Table 13-2 OSB Installation Screens

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you've already downloaded for your organization.
Installation Location	Use this screen to specify the location of your Oracle home directory. If you plan to extend the existing SOA domain, then install the OSB software into the existing Oracle home, where the SOA software has already been installed. If you plan to configure OSB in a separate domain, then install the OSB software in the Infrastructure Oracle home.

Table 13-2 (Cont.) OSB Installation Screens

Screen	Description
Installation Type	Use this screen to select the type of installation and consequently, the products and feature sets that you want to install. For this topology, select Service Bus .
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the following documents in Roadmap for Verifying Your System Environment in <i>Installing and Configuring the Oracle Fusion Middleware Infrastructure</i> .
Installation Summary	Use this screen to verify the installation options that you select. 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 <i>Installing Software with the Oracle Universal Installer</i> . Click Install to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation.
Installation Complete	This screen appears when the installation is complete. Review the information on this screen, then click Finish to dismiss the installer.

Installing the Software on Other Host Computers

If you have configured a separate shared storage volume or partition for SOAHOST2, then you must also install the software on SOAHOST2. For more information, see [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

Note that the location where you install the Oracle home (which contains the software binaries) varies, depending upon the host. To identify the proper location for your Oracle home directories, refer to the guidelines in [File System and Directory Variables Used in This Guide](#).

Validating the OSB Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- [Reviewing the Installation Log Files](#)
- [Checking the Directory Structure](#)
- [Viewing the Contents of Your Oracle Home](#)

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see Understanding Installation Log Files in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

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

The addition of Oracle Service Bus adds the following directory and sub-directories. Use the `ls --format=single-column` command to verify the directory structure:

```
ls --format=single-column $ORACLE_HOME/osb/  
bin  
common  
config  
doc  
financial  
L10N  
lib  
modules  
osb  
plugins  
tools
```

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

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See *Viewing the contents of an Oracle home* in *Installing Software with the Oracle Universal Installer*.

Extending the SOA or Infrastructure Domain to Include Oracle Service Bus

You can use the Configuration Wizard to extend the existing enterprise deployment SOA domain with the Oracle Service Bus. You have to perform a series of additional tasks to complete the extension.

Extending the domain involves the following tasks.

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens to Extend the Domain with Oracle Service Bus](#)

Starting the Configuration Wizard

 **Note:**

SSL store customizations were added to the `setUserOverridesLate.sh` in the domain creation chapter. Any customizations added to this file are preserved when a domain is extended and are carried over to remote servers when using the **pack** and **unpack** commands.

However, if you added any additional customizations to the `setDomainEnv.sh` script in the domain (such as custom libraries, JAVA command line options for starting the servers or environment variables), those will be overwritten by the configuration wizard when you extend the domain. Add all the startup parameters that apply to all servers in a domain to the `setUserOverridesLate.sh` file. This will preserve them across extensions.

To begin domain configuration:

1. Shut down the Administration Server to prevent any configuration locks, saves, or activations from occurring during the configuration of the domain.
2. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

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

Navigating the Configuration Wizard Screens to Extend the Domain with Oracle Service Bus

In this step, you extend the domain created in [Extending the Domain with Oracle SOA Suite](#), and add the Oracle Service Bus software components and Managed Servers.

The steps reflected in this section would be very similar if Oracle Service Bus was extending a domain containing only an Administration Server and a WSM-PM Cluster, but some of the options, libraries, and components shown in the screens could vary.

Follow the instructions in this section to extend the domain for Oracle Service Bus.

 **Note:**

This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks.

- [Task 1, Selecting the Domain Type and Domain Home Location](#)
- [Task 2, Selecting the Configuration Template](#)
- [Task 3, Specifying the Database Configuration Type](#)

- [Task 4, Specifying JDBC Component Schema Information](#)
- [Task 5, Providing the GridLink Oracle RAC Database Connection Details](#)
- [Task 6, Testing the JDBC Connections](#)
- [Task 7, Selecting Advanced Configuration](#)
- [Task 8, Configuring Managed Servers](#)
- [Task 9, Configuring a Cluster](#)
- [Task 10, Assigning Server Templates](#)
- [Task 11, Configuring Dynamic Servers](#)
- [Task 12, Assigning Managed Servers to the Cluster](#)
- [Task 13, Configuring Coherence Clusters](#)
- [Task 14, Verifying the Existing Machines](#)
- [Task 15, Assigning Servers to Machines](#)
- [Task 16, Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Task 17, Writing Down Your Domain Home and Administration Server URL](#)
- [Task 18, Start the Administration Server](#)

Task 1 Selecting the Domain Type and Domain Home Location

On the Configuration Type screen, select **Update an existing domain**.

In the **Domain Location** field, select the value of the `ASERVER_HOME` variable, which represents the complete path to the Administration Server domain home that you created when you created in [Creating the Initial Infrastructure Domain for an Enterprise Deployment](#). For more information about the directory location variables, see [File System and Directory Variables Used in This Guide](#).

For more information about the other options on this screen, see Configuration Type in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 2 Selecting the Configuration Template

On the Templates screen, make sure that **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Service Bus Reference Configuration [osb]**

The following additional templates should already be selected, because they were used to create the initial domain:

- Oracle SOA Suite Reference Configuration [soa] (if you are extending a SOA domain)
- Oracle Enterprise Manager [em]
- Oracle WSM Policy Manager [oracle_common]
- Oracle JRF [oracle_common]
- WebLogic Coherence Cluster Extension [wlserver]

The ODSI XQuery 2004 Components [oracle_common] template is also automatically selected when you select Oracle Service Bus template.

For more information about the options on this screen, see Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

 **Note:**

If you plan to extend the domain to add a component that does not support Reference Configuration, such as BPM and BAM (Reference Configuration is supported only in SOA, OSB, B2B, and ESS), or if you are extending a classic SOA domain with OSB, the OSB classic domain templates must be used.

The classic OSB template, which does not implement the optimizations included in Reference Configuration is as follows:

Oracle Service Bus - 14.1.2.0.0 [osb]

Subsequent extensions on a classic domain for B2B or SOA must be done with Classic extension templates and not with Reference Configuration templates.

Task 3 Specifying the Database Configuration Type

On the Database Configuration Type screen, select **RCU Data**.

All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.

- Verify that the **Vendor** is Oracle and the **Driver** is *Oracle's Driver (Thin) for Service Connections; Versions: Any.
- Verify that **Connection Parameters** is selected.
- Verify and ensure that credentials in all the fields are the same as those provided during the configuration of Oracle Fusion Middleware Infrastructure.

Click **Get RCU Configuration** after you finish verifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK
```

Successfully Done.

 **Tip:**

For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*.

For more information about the other options on this screen, see Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 4 Specifying JDBC Component Schema Information

On the JDBC Component Schema screen, select **OSB JMS Reporting Provider** component schema.

When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.

Click **Convert to GridLink**, and then click **Next**.

Task 5 Providing the GridLink Oracle RAC Database Connection Details

On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, <code>soaedg.example.com</code> .
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	These values are not required when you are using an Oracle 12c database or higher versions because the ONS list is automatically provided from the database to the driver.
Enable Fan	Verify that the Enable Fan check box is selected, so the database can receive and process FAN events.

Task 6 Testing the JDBC Connections

Use the JDBC Component Schema Test screen to test the data source connections that 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.

For more information about the other options on this screen, see Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 7 Selecting Advanced Configuration

To complete domain configuration for the topology, select **Topology** on the Advanced Configuration screen.



Note:

JDBC stores are recommended and selected in [Task 3, Configuring High Availability Options](#) so there is no need to configure File Stores.

If you choose File Stores in [Task 3, Configuring High Availability Options](#), you have to select the File Stores option here to configure them in a shared location in `ORACLE_RUNTIME/domain_name/OSB_Cluster/jms`. Shared location is required to resume JMS and JTA in a failover scenario.

Task 8 Configuring Managed Servers

On the Managed Servers screen, a new Managed Server for Oracle Service Bus appears in the list of servers. This server was created automatically by the Oracle Service Bus configuration template that you selected in [Task 2, Selecting the Configuration Template](#). Perform the following tasks to modify the default Oracle Service Bus Managed Server and create a second Oracle Service Bus Managed Server:

1. Rename the default OSB Managed Server to `WLS_OSB1`.
2. Click **Add** to create a new Managed Server, and name it `WLS_OSB2`.



Tip:

The server names recommended here are used throughout this document; if you choose different names, be sure to replace them as needed.

3. Use the information in [Table 13-4](#) to fill in the rest of the columns for each Managed Server.
4. Select **OSB-MGD-SVRS-ONLY** as the server group for the OSB Servers. Deselect **OSB-MGD-SVRS-COMBINED** that is selected by default.
5. Click **Next**.

For more information about the options on the Managed Server screen, see Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Name	Listen Address	Enable Listen	Listen Port	SSL Listen Port	Enable SSL Port	Administ ration Port	Server Groups
WLS_S OA1	SOAHO ST1	Uncheck ed	Disabled	7004	Checked	9004	SOA- MGD- SVRS- ONLY
WLS_S OA2	SOAHO ST2	Uncheck ed	Disabled	7004	Checked	9004	SOA- MGD- SVRS- ONLY
WLS_W SM1	SOAHO ST1	Uncheck ed	Disabled	7010	Checked	9003	JRF- MAN- SVR WSMPM -MAN- SVR
WLS_W SM2	SOAHO ST2	Uncheck ed	Disabled	7010	Checked	9003	JRF- MAN- SVR WSMPM -MAN- SVR
WLS_O SB1	SOAHO ST1	Uncheck ed	Disabled	8003	Checked	9007	OSB- MGD- SVRS- ONLY
WLS_O SB2	SOAHO ST2	Uncheck ed	Disabled	8003	Checked	9007	OSB- MGD- SVRS- ONLY

The WLS_SOA Managed Servers appear if you extend an existing Oracle SOA Suite domain with Oracle Service Bus.

Task 9 Configuring a Cluster

In this task, you create a cluster of Managed Servers to which you can target the Oracle Service Bus software.

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

For more information about the `osb.example.com` virtual server address, see [Configuring Virtual Hosts on the Hardware Load Balancer](#).

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `OSB_Cluster` in the **Cluster Name** field.
3. Specify `osb.example.com` in the **Frontend Host** field.
4. Specify 0 as the **Frontend HTTP Port** and 443 as the **Frontend HTTPS port**.



Note:

By default, server instances in a cluster communicate with one another by 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*.

For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 10 Assigning Server Templates

Click **Next** to continue.

Task 11 Configuring Dynamic Servers

Click **Next** to continue.

Task 12 Assigning Managed Servers to the Cluster

On the Assign Servers to Clusters screen, assign servers to clusters as follows:

Note that the WLS_SOA Managed Servers appear only if you extend an existing Oracle SOA Suite domain with Oracle Service Bus.

- SOA_Cluster (If you are extending a SOA domain):
 - WLS_SOA1
 - WLS_SOA2
- WSM-PM_Cluster:
 - WLS_WSM1
 - WLS_WSM2
- OSB_Cluster:
 - WLS_OSB1
 - WLS_OSB2

Click **Next**.

For more information about the options on this screen, see Assign Servers to Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 13 Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at 9991, as it was defined during the initial Infrastructure domain creation.

For Coherence licensing information, see Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 14 Verifying the Existing Machines

Confirm that the following entries appear:

Name	Node Manager Listen Address
SOAHOST1	SOAHOST1
SOAHOST2	SOAHOST2
ADMINHOST	ADMINVHN

Leave all other fields to their default values.
Click **Next**.

Task 15 Assigning Servers to Machines

On the Assign Servers to Machines screen, assign servers to machines as follows:

- ADMINHOST:
 - AdminServer
- SOAHOST1
 - WLS_SOA1 (if extending a SOA domain)
 - WLS_WSM1
 - WLS_OSB1
- SOAHOST2:
 - WLS_SOA2 (if extending a SOA domain)
 - WLS_WSM2
 - WLS_OSB2

For more information about the options on this screen, see Assign Servers to Machines in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 16 Reviewing Your Configuration Specifications and Configuring the Domain

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

Click **Update** to execute the domain extension.

In the Configuration Progress screen, click **Next** when it finishes.

For more information about the options on this screen, see Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 17 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen shows the following items about the domain you just configured, including:

- Domain Location

- Administration Server URL

Make a note of both these items, because you need them later; you need the domain location to access the scripts used to start the Administration Server, and you need the Administration Server URL to access the WebLogic Remote Console and Oracle Enterprise Manager Fusion Middleware Control.

Click **Finish** to dismiss the Configuration Wizard.

Task 18 Start the Administration Server

Start the Administration Server to ensure the changes that you have made to the domain have been applied.

After you have completed extending the domain, go to [Propagating the Extended Domain to the Domain Directories and Machines](#).

Update Certificates for New Frontend Addresses

This section contains information about certificates for new frontend addresses.

Note:

About Certificates for the domain extension. Since the OSB and WSM servers use the same listen addresses (different port), there is no need to create new certificates and update stores. However, the OSB cluster uses a different front end address that will be added as a trusted endpoint in the [Configuring the Web Tier for the Extended Domain](#).

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in the [Updating the WebLogic Servers Security Settings](#) and update SSL settings for the WLS_OSB1 and WLS_OSB2 servers.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the OSB instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and systems.

Note that there is no need to propagate the updated domain to the WEBHOST1 and WEBHOST2 systems, because there are no changes to the Oracle HTTP Server instances on those host computers.

Refer to the following sections for more information.

- [Summary of the Tasks Required to Propagate the Changes to the Other Domain Directories and Machines](#)
- [Starting and Validating the WLS_OSB1 Managed Server](#)
- [Starting and Validating the WLS_OSB2 Managed Server](#)

- [Verifying the Appropriate Targeting and Configuration for OSB Singleton Services](#)

Summary of the Tasks Required to Propagate the Changes to the Other Domain Directories and Machines

[Table 13-5](#) summarizes the steps required to propagate the changes to all the domain directories and systems.

Table 13-5 Summary of Tasks Required to Propagate the Domain Chanegs to Domain Directories and Machines

Task	Description	More Information
Pack up the Extended Domain on SOAHOST1	Use the <code>pack</code> command to create a new template jar file that contains the new OSB Servers configuration. When you pack up the domain, create a template jar file called <code>soadomaintemplateExtOSB.jar</code> .	Packing Up the Extended Domain on SOAHOST1
Unpack the Domain in the Managed Servers Directory on SOAHOST1	Unpack the template jar file in the Managed Servers directory on SOAHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1
Unpack the Domain on SOAHOST2	Unpack the template jar file in the Managed Servers directory on the SOAHOST2 local storage.	Unpacking the Domain on SOAHOST2

Starting and Validating the WLS_OSB1 Managed Server

After you extend the domain, restart the Administration Server, and propagate the domain to the other hosts, use the following procedure to start the WLS_OSB1 server and validate if the server is configured successfully:

- [Starting the WLS_OSB1 Managed Server](#)
- [Adding the MiddlewareAdministrators Role to the Enterprise Deployment Administration Group](#)
- [Validating the Managed Server](#)

Starting the WLS_OSB1 Managed Server

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

`https://ADMINVHN:9002/em`

Note:

If you have already configured web tier, use `https://admin.example.com:445/em`.

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

3. In the **Target Navigation** pane, expand the domain to view the Managed Servers in the domain.
4. Select only the **WLS_OSB1** Managed Server, and click **Start Up** on the Oracle WebLogic Server toolbar.

 **Note:**

OSB Servers depend on the policy access service to be functional. This implies that the WSM-PM Managed Servers in the domain need to be up and running and reachable before the OSB servers are started.

5. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_OSB1 Managed Server is up and running.

Adding the MiddlewareAdministrators Role to the Enterprise Deployment Administration Group

Before you validate the Oracle Service Bus configuration on the WLS_OSB1 Managed Server, add the Oracle Service Bus `MiddlewareAdministrators` administration role to the enterprise deployment administration group (`SOA Administrators`) and add the `IntegrationAdministrators` group in the external LDAP directory.

To perform this task, refer to [Configuring Roles for Administration of Oracle SOA Suite Products](#).

Validating the Managed Server

After you add the `MiddlewareAdministrator` role to the `SOA Administrators` group, you can validate the configuration of the Oracle Service Bus software on the WLS_OSB1 Managed Server as follows:

1. Use your web browser to navigate to the following URL:

```
https://SOAHOST1:8003/sbinspection.wsil
```

Replace `SOAHOST1` with the value of this variable in the *Enterprise Deployment Workbook*. For more information about the physical IP (IP) and virtual IP (VIP) addresses required for the Administration server and each of the managed servers, see [Physical and Virtual IP Addresses Required by the Enterprise Topology](#).

2. Log in by using the enterprise deployment administration user (`SOA Administrators`).

With the default installation, this should result in the following HTTP response to the Web services call:

```
<ins:inspection xmlns:ins="http://schemas.xmlsoap.org/ws/2001/10/inspection/">
```

Starting and Validating the WLS_OSB2 Managed Server

Follow similar steps as in the previous section for WLS_OSB2:

1. Log in to Fusion Middleware Control by using the Administration Server credentials.
2. In the Target Navigation pane, expand the domain to view the Managed Servers in the domain.

3. Select only the WLS_OSB2 Managed Server, and click **Start Up** on the Oracle WebLogic Server tool bar.
4. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_OSB2 Managed Server is up and running. Access the equivalent URLs for the WLS_OSB2:

`https://SOAHOST2:8003/sbinspection.wsil`

5. Verify the correct deployment of the Oracle Service Bus console to the Administration Server by accessing the following URL:

`https://ADMINVHN:9002/servicebus/`

Verifying the Appropriate Targeting and Configuration for OSB Singleton Services

Oracle Service Bus uses some singleton services that should run only in one of the WLS servers in the OSB_Cluster. These singleton services are:

- Aggregator
- SLA Alert Manager
- Poller transports (Email, File, FTP, and SFTP pollers)

This is controlled by the global property **OSB Singleton Components Automatic Migration**, which is exposed in Enterprise Manager, in the Global Settings section of the Service Bus configuration. When activated, it uses the WebLogic Singleton Framework to guarantee the singleton behavior and automatic migration of these OSB singleton services.

Similar to server or service migration, a database leasing is a requirement for the OSB Singleton Components Automatic Migration to work properly. The **OSB Singleton Components Automatic Migration** checkbox does not automatically define the leasing datasource, it just marks these applications as singletons.

This OSB global property and the Database Leasing are checked by default for the SOA Enterprise Deployment topologies as long as **Enable Automatic Service Migration** was selected during the configuration wizard as recommended in this guide.

For cases where Automatic Service Migration was not enabled using the Configuration Wizard, and you define it manually afterwards, you must also check OSB Singleton Components Automatic Migration manually.

To guarantee the appropriate Singleton behavior for OSB:

Verify that the **OSB Singleton Components Automatic Migration** option is checked:

1. Log in to Oracle Fusion Middleware Enterprise Manager. In a browser, go to the following URL:

`https://ADMINVHN:9002/em`

2. Navigate to **SOA > service-bus (AdminServer) > Global Settings**.

The property **OSB Singleton Components Automatic Migration** must be checked by default for the SOA EDG topologies.

Verify that the appropriate targeting exists by following these steps:

1. Log into the WebLogic Remote Console.
2. In the **Edit Tree**, and then click **App Deployments**.

3. Find the **Aggregator Singleton Marker Application**. Verify that the value in the Targets column of the table is **OSB_Cluster**.

Verify that the leasing datasource is defined for the OSB_Cluster:

1. Log into the WebLogic Remote Console.
2. In the **Edit Tree**, expand **Environment**, and then click **Clusters**.
3. Click **OSB_Cluster**.
4. Click the **Migration** tab.
5. Verify that **Database** is selected in the **Migration Basis** drop-down menu and the leasing datasource **Data Source For Automatic Migration** is defined.

If database leasing is not defined, see [Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster](#) for instructions to configure it.

Note:

It is assumed that at the time of configuring the domain with the Config Wizard, the **Enable Automatic Service Migration** option is checked in the High Availability Options screen. If the option is not checked, **Service Bus Domain Singleton Marker Application** is targeted directly to the first server of the cluster **WLS_OSB1** and this server hosts the singleton services. Oracle does not recommend this approach because it does not provide automatic migration of the OSB singletons services in case of a failure.

Modifying the Upload and Stage Directories to an Absolute Path

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. See [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).

Configuring the Web Tier for the Extended Domain

It is important to understand how to configure the web server instances on the web tier so that they route requests for both public and internal URLs to the proper clusters in the extended domain.

- [Generate the Required Certificates for OHS SSL Listeners](#)
- [Configuring Oracle HTTP Server for the Oracle Service Bus](#)
Oracle Service Bus exposes applications deployed both in the Administration Server and in the Oracle Service Bus Cluster. The Oracle HTTP Server mount points required by Oracle Service Bus in the Administration server are added to the `admin_vh.conf` file created in the domain creation chapter. For the application mounts in the Oracle Service Bus cluster you can reuse your SOA frontend virtual hostname (`soa.example.com`) or use an specific virtual hostname for Oracle Service Bus (such as `osb.example.com`, reflected in the topology diagrams).

- [Validating the Oracle Service Bus URLs Through the Load Balancer](#)
Verify the Oracle Service Bus URLs to ensure that appropriate routing and failover is working from the hardware load balancer to the HTTP Server instances to the Oracle Service Bus components.

Generate the Required Certificates for OHS SSL Listeners

Follow the steps described in the [Generate Required Certificates for OHS SSL Listeners](#) section in [Starting the Oracle HTTP Server Instances](#) to add the new fronted address to the certificate stores and update the SAN for the OHS listeners certs.

When asked to replace the existing OHS Virtual Host certificates, answer yes so that they are updated with the new frontend address for the OSB cluster as SAN.

Configuring Oracle HTTP Server for the Oracle Service Bus

Oracle Service Bus exposes applications deployed both in the Administration Server and in the Oracle Service Bus Cluster. The Oracle HTTP Server mount points required by Oracle Service Bus in the Administration server are added to the `admin_vh.conf` file created in the domain creation chapter. For the application mounts in the Oracle Service Bus cluster you can reuse your SOA frontend virtual hostname (`soa.example.com`) or use an specific virtual hostname for Oracle Service Bus (such as `osb.example.com`, reflected in the topology diagrams).

In the first case you must add the mount points required by OSB to expose the pertaining applications with the existing OHS virtual host `soa_vh.conf` configuration file. In the second case you need to perform the following steps:

1. Configure your load balancer to add an additional Virtual Hostname/Virtual Server for `osb.example.com`. For more information, see your precise Load Balancer vendor documentation.
2. Configure the OHS instances to serve the new virtual hostname (`osb.example.com`).

Perform the following steps to configure the OHS instances to serve the new virtual hostname:

1. Repeat the steps described in the [Generate Required Certificates for OHS SSL Listeners](#) section in [Configuring Oracle HTTP Server for an Enterprise Deployment](#) to update the certificate stores with the new virtual server (`osb.example.com`).
2. Log into WEBHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (ohs1):

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

3. Copy the existing `admin_vh.conf` file to the `osb_vh.conf` file. This will transfer most of the required SSL configuration:

```
cp admin_vh.conf osb_vh.conf
```

4. Edit the file and customize with the required values for **Listen**, **ServerName**, **VirtualHost**, **SSLWallet** and **Location** directives (AllowEncodedSlashes not needed here):

 **Note:**

For the Listen address, you need to specify a different port from the ones used in previous virtual hosts (admin_vh.conf , soainternal_vh.conf or soa_vh.conf). Otherwise the listeners will conflict.

```
#####
# Oracle HTTP Server mod_ossll configuration file: ssl.conf      #
#####

# The Listen directive below has a comment preceding it that is used
# by tooling which updates the configuration. Do not delete the comment.
#[Listen] OHS_SSL_PORT
Listen WEBHOST1:4446
##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost WEBHOST1:4446>
ServerName osb.example.com:443

<IfModule ossl_module>

# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# Client Authentication (Type):
# Client certificate verification type and depth. Types are
# none, optional and require.
SSLVerifyClient None

# SSL Protocol Support:
# Configure usable SSL/TLS protocol versions.
SSLProtocol TLSv1.2 TLSv1.3

# Option to prefer the server's cipher preference order
SSLHonorCipherOrder on

# SSL Cipher Suite:
# List the ciphers that the client is permitted to negotiate.
SSLCipherSuite
TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_S
A384,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_S
HA384

#Path to the wallet
#SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/${
{COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/default"
SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1
```

```

<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>

<Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/${
COMPONENT_TYPE}/instances/${COMPONENT_NAME}/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

# Add the following directive to add HSTS
<IfModule mod_headers.c>
    Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
</IfModule>

<Location /sbinspection.wsil>
    WLSRequest ON
    WebLogicCluster SOAHOST1:8003,SOAHOST2:8003
</Location>

<Location /sbresource>
    WLSRequest ON
    WebLogicCluster SOAHOST1:8003,SOAHOST2:8003
</Location>

<Location /osb>
    WLSRequest ON
    WebLogicCluster SOAHOST1:8003,SOAHOST2:8003
</Location>

<Location /alsb>
    WLSRequest ON
    WebLogicCluster SOAHOST1:8003,SOAHOST2:8003
</Location>

<Location /default>
    WLSRequest ON
    WebLogicCluster SOAHOST1:8003,SOAHOST2:8003
</Location>

</IfModule>
</VirtualHost>

```

5. Add the following entry to the `admin_vh.conf` file within the `<VirtualHost>` tags:

```

<Location /servicebus>
    WLSRequest ON
    WebLogicHost ADMINVHN
    WeblogicPort 9002
</Location>

```

```
<Location /testconsole >  
  WLSRequest ON  
  WebLogicHost ADMINVHN  
  WeblogicPort 9002  
</Location>
```

6. Log into WEBHOST2 and copy the `osb_vh.conf` file and the `admin_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (ohs2):

```
$WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf
```

7. Edit the `osb_vh.conf` file and change any references to WEBHOST1 to WEBHOST2 in the `<VirtualHost>` directives.
8. Restart Oracle HTTP Servers on WEBHOST1 and WEBHOST2.

Validating the Oracle Service Bus URLs Through the Load Balancer

Verify the Oracle Service Bus URLs to ensure that appropriate routing and failover is working from the hardware load balancer to the HTTP Server instances to the Oracle Service Bus components.

To verify the URLs:

1. While WLS_OSB1 is running, stop WLS_OSB2 by using the Oracle WebLogic Remote Console.
2. Access the following URL and verify the HTTP response as indicated in [Starting and Validating the WLS_OSB2 Managed Server](#):

```
https://osb.example.com/sbinspection.wsil
```

3. Start WLS_OSB2 from the Oracle WebLogic Remote Console.
4. Stop WLS_OSB1 from the Oracle WebLogic Remote Console.
5. Access the same URL and verify the HTTP response as indicated in [Starting and Validating the WLS_OSB2 Managed Server](#).

Note:

Since a front end URL has been set for the OSB_Cluster, the requests to the urls result in a reroute to the LBR, but in all cases it should suffice to verify the appropriate mount points and correct failover in Oracle HTTP Server.

6. Verify this URL by using your load balancer address:

```
https://osb.example.com:443/sbinspection.wsil
```

You can also verify `https://admin.example.com:445/servicebus`.

Post-Configuration Tasks for Oracle Service Bus

After you install and configure Oracle Service Bus in the domain, consider the following post-configuration tasks.

- [Enabling High Availability for Oracle DB, File and FTP Adapters](#)
- [Considerations for Poller Transports](#)
OSB provides native Poller transports that are not transactional in nature. These transports poll a source directory, FTP server, or email server for new messages and push the processing payloads to the required JMS destinations. Email, File, FTP, and SFTP fall in this category.
- [Configuring Specific Oracle Service Bus Services for an Enterprise Deployment](#)

Enabling High Availability for Oracle DB, File and FTP Adapters

Oracle SOA Suite and Oracle Service Bus use the same database, file, and FTP JCA adapters.

You create the required database schemas for these adapters when you use the Oracle Repository Creation Utility before you configure Oracle SOA Suite. The database adapter does not require any configuration at the WebLogic Server resource level.

The required configuration for the other adapters is described in section [Enabling High Availability for Oracle File and FTP Adapters](#).

If you configure Oracle Service Bus as an extension of a SOA domain, you do not need to add to the configuration already performed for the adapters.

If you deploy Oracle Service Bus as an extension to an Oracle Fusion Middleware Infrastructure domain (without Oracle SOA Suite), perform the steps as described in [Enabling High Availability for Oracle File and FTP Adapters](#).

Considerations for Poller Transports

OSB provides native Poller transports that are not transactional in nature. These transports poll a source directory, FTP server, or email server for new messages and push the processing payloads to the required JMS destinations. Email, File, FTP, and SFTP fall in this category.

Poll-based transports use a transport poller thread that is pinned to a Managed Server. All Managed Servers in a cluster can process the pertaining payload, but only one server can poll for the message. To protect the system from outages, the poller thread must be configured as an application-scoped singleton and the involved JMS destinations must be highly available.

The Poller Transport singleton behavior, similar to the other OSB singleton services, is controlled by the property **OSB Singleton Components Automatic Migration**.

When checked, an application-scoped singleton for the poller is deployed to the cluster. Similar to server or service migration, a leasing is a requirement for the **OSB Singleton Components Automatic Migration** to work properly.

**Note:**

This checkbox does not automatically define the leasing datasource, it just marks the applications as singletons.

The property OSB Singleton Components Automatic Migration and the Database Leasing are checked by default for the SOA Enterprise Deployment topologies, if **Enable Automatic Service Migration** was selected in the Configuration Wizard, as recommended in this guide.

If the leasing datasource is not already configured and this checkbox is activated, ensure that you configure the leasing datasource. See [Verifying the Appropriate Targeting and Configuration for OSB Singleton Services](#).

You can verify that your poller transport is configured as an application-scoped singleton for OSB Singleton Components Automatic Migration by following these steps:

1. Log into the WebLogic Remote Console.
2. In the **Edit Tree**, click **Deployments**.
3. In the **Domain Structure** tree on the left, click **Deployments**.
4. Verify that there is a singleton application deployed for the transport poller. Example: SB_FILE_Proxy_*
5. Verify that the value in the **Targets** column of the table is **OSB_Cluster**.

For the high availability of this transport services, it is required to protect the pertaining JMS destinations (used for payload processing) from failures. In SOA Enterprise Deployment topology, this protection is provided by the Automatic Service Migration feature.

Configuring Specific Oracle Service Bus Services for an Enterprise Deployment

To use IBM WebSphere MQ Connection resources and the MQ Transport in Oracle Service Bus, you must add the MQ client libraries to the classpath.

One option is to copy the required MQ libraries to the following location in the domain home directory:

```
DOMAIN_HOME/lib
```

This is also the case for custom assertions and JBoss integration services:

- When you use JBoss initial context factory classes, make sure to include the class and any dependent classes in the `DOMAIN_HOME/lib` directory.
- Similarly, for custom assertions, create the required jar file with the assertion and add the jar to the `DOMAIN_HOME/lib` directory.

Further, to use these services in an enterprise deployment, you must add the required libraries to the Administration Server domain home (`ASERVER_HOME/lib`) and the Managed Server domain home (`MSERVER_HOME/lib`).

For more information about configuring and developing services for Oracle Service Bus, see *Getting Started with the Oracle Service Bus Console* in *Developing Services with Oracle Service Bus*.

Replacing Connect Strings with the Appropriate TNS Alias

Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

For more information about how to use TNS alias in your Datasources, see [Using TNS Alias in Connect Strings](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see [Performing Backups and Recoveries for an Enterprise Deployment](#).

Extending the Domain with Business Process Management

The procedures described in this chapter guide you through the process of extending the enterprise deployment topology to include Business Process Management (BPM).

- [Variables Used When Configuring Business Process Management](#)
As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.
- [Support for Reference Configuration in Business Process Management](#)
Oracle BPM does not support the Reference Configuration. Hence, you can extend BPM only with a Classic SOA domain that has been created using the classic SOA domain templates.
- [Prerequisites for Extending the SOA Domain to Include Oracle BPM](#)
Before you extend the current domain, ensure that your existing deployment meets the necessary prerequisites.
- [Installing Oracle Business Process Management for an Enterprise Deployment](#)
The installation of Oracle SOA Foundation and Business Process Management software for an enterprise deployment is a three-step process.
- [Running the Configuration Wizard on SOAHOST1 to Extend a SOA Domain to Include BPM](#)
Run the Configuration Wizard from the `ORACLE_COMMON_HOME` directory to extend a domain that contains an Administration Server, Oracle Web Services Manager and SOA to support BPM components.
- [Propagating the Extended Domain to the Domain Directories and Machines](#)
Oracle BPM Suite requires some updates to the WebLogic Server start scripts. Propagate these changes by using the `pack` and `unpack` commands.
- [Starting the WLS_SOA Managed Servers with Business Process Management](#)
For configuration changes and start scripts to be effective, you must start the `WLS_SOA`n server to which BPM has been added.
- [Adding the Enterprise Deployment Administration User to the Oracle BPM Administrators Group](#)
Before you validate the Oracle Business Process Management configuration on the Managed Server, add the enterprise deployment administration user (`weblogic_soa`) to the Business Process Management `Administrators` group in the LDAP directory.
- [Configuring the Web Tier for the Extended Domain](#)
Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.
- [Validating Access to Business Process Management Through the Hardware Load Balancer](#)
Because the cluster address for the `SOA_Cluster` has already been set in the previous chapter, the Business Process Management system can be verified only after the Oracle HTTP Server configuration files have been modified to route the Business Process Management context URLs to the WebLogic Servers.

- [Configuring BPMJMSModule for the Oracle BPM Cluster](#)
When you configure Oracle Business Process Management in a Oracle WebLogic Server domain, the BPMJMSModule JMS module is deployed automatically.
- [Replacing Connect Strings with the Appropriate TNS Alias](#)
Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.
- [Backing Up the Configuration](#)
It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

Variables Used When Configuring Business Process Management

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

The values for several directory variables are defined in [File System and Directory Variables Used in This Guide](#).

- ORACLE_HOME
- ASERVER_HOME
- MSERVER_HOME
- WEB_DOMAIN_HOME
- JAVA_HOME

In addition, you reference the following virtual IP (VIP) addresses that are defined in [Physical and Virtual IP Addresses Required by the Enterprise Topology](#):

- ADMINVHN

Actions in this chapter are performed on the following host computers:

- SOAHOST1
- SOAHOST2
- WEBHOST1
- WEBHOST2

Support for Reference Configuration in Business Process Management

Oracle BPM does not support the Reference Configuration. Hence, you can extend BPM only with a Classic SOA domain that has been created using the classic SOA domain templates.

Prerequisites for Extending the SOA Domain to Include Oracle BPM

Before you extend the current domain, ensure that your existing deployment meets the necessary prerequisites.

- Back up the installation. If you have not yet backed up the existing Fusion Middleware Home and domain, Oracle recommends backing it up now.

To back up the existing Fusion Middleware Home and domain, see [Performing Backups and Recoveries in the SOA Enterprise Deployments](#).

- Existing `WL_HOME` and `SOA_ORACLE_HOME` (binaries) are installed in previous chapters on a shared storage and are available from `SOAHOST1` and `SOAHOST2`.
- Node Manager, Admin Server, SOA Servers and WSM Servers exist and have been configured as described in previous chapters to run a SOA system.
- You do not need to run RCU to load additional schemas for BPM. These are part of the SOA repository and are loaded into the DB in the SOA chapter

Installing Oracle Business Process Management for an Enterprise Deployment

The installation of Oracle SOA Foundation and Business Process Management software for an enterprise deployment is a three-step process.

- [Starting the Installation Program](#)
- [Navigating the Oracle BPM Installation Screens](#)
- [Installing the Software on Other Host Computers](#)
- [Verifying the Installation](#)

Starting the Installation Program

To start the installation program, perform the following steps.

1. Log in to the target system.
2. Make sure that a certified JDK already exists on your system. See [Installing a Supported JDK](#)
3. Go to the directory where you downloaded the installation program.
4. Launch the installation program by running the `java` executable from the JDK directory on your system, as shown in the example below.

```
$JAVA_HOME/bin/java -jar fmw_14.1.2.0.0_soa.jar
```

See [Identifying and Obtaining Software Distributions for an Enterprise Deployment](#).

Be sure to replace JDK location in these examples with the actual JDK location on your system.

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

Navigating the Oracle BPM Installation Screens

The installation program displays a series of screens, in the order listed in [Table 14-1](#).

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

Table 14-1 Oracle Business Process Management Install Screens



Screen	Description
Installation Inventory Setup	<p>On UNIX operating systems, if this is the first time that you are installing any Oracle product on this host, this screen appears. 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.</p> <p>For more information about the central inventory, see <i>Understanding the Oracle Central Inventory</i> in <i>Installing Software with the Oracle Universal Installer</i>.</p> <div style="border: 1px solid #0070C0; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>Oracle recommends that you configure the central inventory directory on the products shared volume.</p> <p>Example: <code>/u01/oracle/products/oraInventory</code></p> <p>You may also need to execute the <code>createCentralInventory.sh</code> script as root from the <code>oraInventory</code> folder after the installer completes.</p> </div>
Auto Updates	<p>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.</p>
Installation Location	<p>Use this screen to specify the location of your Oracle home directory. For the Oracle Home, specify <code>/u01/oracle/products/fmwnnnn</code>.</p> <p>For more information about Oracle Fusion Middleware directory structure, see <i>Selecting Directories for Installation and Configuration</i> in <i>Planning an Installation of Oracle Fusion Middleware</i>.</p>

Table 14-1 (Cont.) Oracle Business Process Management Install Screens

Screen	Description
Installation Type	<p>Use this screen to select the type of installation and consequently, the products and feature sets that you want to install.</p> <ul style="list-style-type: none"> Select BPM
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>The topology in this document does not include the examples, Oracle strongly recommends that you do not install the examples into a production environment.</p> </div>
Prerequisite Checks	<p>This screen verifies that your system meets the minimum necessary requirements.</p> <p>If there are any warning or error messages, you can refer to one of the following documents in Roadmap for Verifying Your System Environment in <i>Installing and Configuring the Oracle Fusion Middleware Infrastructure</i>.</p>
Installation Summary	<p>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.</p> <p>For more information about silent or command-line installation, see Using the Oracle Universal Installer in Silent Mode in <i>Installing Software with the Oracle Universal Installer</i>.</p> <p>Click Install to begin the installation.</p>
Installation Progress	<p>This screen allows you to see the progress of the installation. Click Next when the progress bar reaches 100% complete.</p>
Installation Complete	<p>Review the information on this screen, then click Finish to dismiss the installer.</p>

Installing the Software on Other Host Computers

If you have configured a separate shared storage volume or partition for SOAHOST2, then you must also install the software on SOAHOST2. For more information, see [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

Note that the location where you install the Oracle home (which contains the software binaries) varies, depending upon the host. To identify the proper location for your Oracle home directories, refer to the guidelines in [File System and Directory Variables Used in This Guide](#).

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- [Reviewing the Installation Log Files](#)
- [Checking the Directory Structure](#)
- [Viewing the Contents of Your Oracle Home](#)

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see *Understanding Installation Log Files* in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure

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

The addition of BPM adds the following directory and sub-directories to the `ORACLE_HOME/soa/bpm` directory. Use the `ls --format=single-column` command to verify the list of directories:

```
ls --format=single-column $ORACLE_HOME/soa/bpm
composites
helpsets
lib
modules
projects
```

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

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See *Viewing the contents of an Oracle home* in *Installing Software with the Oracle Universal Installer*.

Running the Configuration Wizard on SOAHOST1 to Extend a SOA Domain to Include BPM

Run the Configuration Wizard from the `ORACLE_COMMON_HOME` directory to extend a domain that contains an Administration Server, Oracle Web Services Manager and SOA to support BPM components.

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens to Extend the Domain with BPM](#)

Starting the Configuration Wizard

 **Note:**

SSL store customizations were added to the `setUserOverridesLate.sh` in the domain creation chapter. Any customizations added to this file are preserved when a domain is extended and are carried over to remote servers when using the **pack** and **unpack** commands.

However, if you added any additional customizations to the `setDomainEnv.sh` script in the domain (such as custom libraries, JAVA command line options for starting the servers or environment variables), those will be overwritten by the configuration wizard when you extend the domain. Add all the startup parameters that apply to all servers in a domain to the `setUserOverridesLate.sh` file. This will preserve them across extensions.

To start the Configuration Wizard:

1. Stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.

 **Note:**

This specific domain extension for Oracle Business Process Management component modifies the WLS_SOAn Managed Servers. Be sure to shut down these Managed Servers.

2. Verify the status of the Managed Servers, and then stop the Administration Server.
3. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

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

Navigating the Configuration Wizard Screens to Extend the Domain with BPM

In this step, you extend the domain created in [Extending the Domain with Oracle SOA Suite](#) and add the BPM components to the Managed Servers. Follow the instructions in these sections to extend the domain for BPM.

 **Note:**

This procedure assumes you are extending an existing domain. If your needs do not match the instructions given in the procedure, be sure to make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- [Task 1, Selecting the Domain Type and Domain Home Location](#)
- [Task 2, Selecting the Configuration Template](#)
- [Task 3, Providing the GridLink Oracle RAC Database Connection Details](#)
- [Task 4, Testing the JDBC Connections](#)
- [Task 5, Selecting Advanced Configuration](#)
- [Task 6, Reviewing your Configuration Specifications and Configuring the Domain](#)
- [Task 7, Writing Down Your Domain Home and Administration Server URL](#)
- [Task 8, Start the Administration Server](#)

Task 1 Selecting the Domain Type and Domain Home Location

On the Configuration Type screen, select Update an existing domain.

In the Domain Location field, select the value of the `ASERVER_HOME` variable, which represents the complete path to the Administration Server domain home you created in [Creating the Initial Infrastructure Domain for an Enterprise Deployment](#).

For more information about the directory location variables, see [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 Template

On the Templates screen, make sure Update Domain Using Product Templates is selected, then select the following templates:

- **Oracle BPM Suite - 14.1.2.0.0 [soa]**

In addition, the following additional templates should already be selected, because they were used to create the initial domain and extend it to SOA:

- Basic Weblogic Server Domain - 14.1.2.0.0 [wlserver]
- Oracle SOA Suite - 14.1.2.0.0 [soa]
- Oracle Enterprise Manager - 14.1.2.0.0 [em]
- Oracle WSM Policy Manager - 14.1.2.0.0 [oracle_common]
- Oracle JRF - 14.1.2.0.0 [oracle_common]
- WebLogic Coherence Cluster Extension - 14.1.2.0.0 [wlserver]
- Oracle Service Bus - 14.1.2.0.0 [osb] (if the domain was extended with Oracle Service Bus)

Tip:

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

Task 3 Providing the GridLink Oracle RAC Database Connection Details

All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain. BPM uses the existing Datasources for SOA and no new Datasources need to be added to the domain.

Note:

Any custom datasources that were created before the extension (such as LEASING datasources) will show up before this screen. Check the Datasources row and click **Next**. The test datasource screen will verify its validity. Click **Next**.

Task 4 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.

For more information about the other options on this screen, see Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 5 Selecting Advanced Configuration

To complete domain configuration for the topology, do not select any additional options on the Advanced Configuration screen. Click **Next**. BPM applications and required artifacts will be targeted automatically to the existing SOA servers.

Task 6 Reviewing your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the detailed configuration information for the domain you are about to extend. Review the details of each item on the screen and verify that the information is correct.

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.

Click **Update** to execute the domain extension.

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 7 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 Administration Server, and the URL is needed to access the Administration Server.

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

Task 8 Start the Administration Server

If the Admin Server was running during the domain extension process, restart the server to ensure the changes you have made to the domain have been applied.

Propagating the Extended Domain to the Domain Directories and Machines

Oracle BPM Suite requires some updates to the WebLogic Server start scripts. Propagate these changes by using the `pack` and `unpack` commands.

[Table 14-2](#) summarizes the steps that are required to propagate the changes to all the domain directories and systems.

Note that there is no need to propagate the updated domain to the WEBHOST1 and WEBHOST2 machines, because there are no changes to the Oracle HTTP Server instances on those host computers.

Table 14-2 Summary of Tasks Required to Propagate the Domain Changes to Domain Directories and Machines

Task	Description	More Information
Pack up the Extended Domain on SOAHOST1	Use the Pack command to create a new template jar file that contains the new Oracle BPM Suite Managed Servers configuration. When you pack up the domain, create a template jar file called <code>soadomaintemplateExtSOABPM.jar</code> .	Packing Up the Extended Domain on SOAHOST1
Unpack the Domain in the Managed Servers Directory on SOAHOST1	Unpack the template jar file in the Managed Servers directory on SOAHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1
Unpack the Domain on SOAHOST2	Unpack the template jar file in the Managed Servers directory on the SOAHOST2 local storage.	Unpacking the Domain on SOAHOST2

Starting the WLS_SOA Managed Servers with Business Process Management

For configuration changes and start scripts to be effective, you must start the `WLS_SOAn` server to which BPM has been added.

Because BPM extends an already existing SOA system, the Administration Server and respective Node Managers are already running in SOAHOST1 and SOAHOST2.

To start the `WLS_SOA1` Managed Server:

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

`https://ADMINVHN:9002/em`
2. Log in to Fusion Middleware Control by using the Administration Server credentials.

3. In the **Target Navigation** pane, expand the domain to view the Managed Servers in the domain.
4. Ensure that the server is still selected and click **Start Up** in the toolbar.
5. Repeat steps 3 and 4 for the WLS_SOA2 Managed Server.
6. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_SOA1 and WLS_SOA2 Managed Servers are up and running.

Adding the Enterprise Deployment Administration User to the Oracle BPM Administrators Group

Before you validate the Oracle Business Process Management configuration on the Managed Server, add the enterprise deployment administration user (`weblogic_soa`) to the Business Process Management `Administrators` group in the LDAP directory.

To perform this task, refer to [Configuring Roles for Administration of Oracle SOA Suite Products](#).

Note that the first time you log in to the Business Process Management Composer or Business Process Management Worklist applications, you must log in as a user that is a member of the Administrators group. After the initial login, any user can be an administration user, as long as they are granted the following roles:

`OracleBPMComposerRolesApp/BPMComposerAdmin`

Also, after the first login, any authenticated user should be able to access the Business Process Management applications.

Configuring the Web Tier for the Extended Domain

Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.

- [Configuring Oracle HTTP Server for Oracle Business Process Management](#)
Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle Business Process Management requests correctly to the Oracle Business Process Management software.

Configuring Oracle HTTP Server for Oracle Business Process Management

Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle Business Process Management requests correctly to the Oracle Business Process Management software.

To enable Oracle HTTP Server to route requests to the BPM Composer and BPM Workspace console:

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

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

2. Add the following directives inside the `<VirtualHost>` tag in the `soa_vh.conf` file:

 **Note:**

The `WebLogicCluster` directive needs only a sufficient number of redundant `server:port` combinations to guarantee an initial contact in case of a partial outage. The actual total list of cluster members is retrieved automatically on the first contact with any given node.

```
# BPM
<Location /bpm/composer>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# BPM
<Location /bpm/workspace>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# To upload attachments
<Location /bpm/casemgmt>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

<Location /frevvo>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>
```

3. Log in to `WEBHOST2` and change the directory to the following location so that you can update the configuration file for the second Oracle HTTP Server instance (`ohs2`):

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf
```

4. Open the `soa_vh.conf` file and add the Oracle Business Process Management directives to the `<VirtualHost>` tag.
5. Restart Oracle HTTP Servers on `WEBHOST1` and `WEBHOST2`.

Validating Access to Business Process Management Through the Hardware Load Balancer

Because the cluster address for the `SOA_Cluster` has already been set in the previous chapter, the Business Process Management system can be verified only after the Oracle HTTP Server configuration files have been modified to route the Business Process Management context URLs to the WebLogic Servers.

Use the following procedure to verify the Business Process Management URLs to ensure that appropriate routing and failover is working from the hardware load balancer to the Oracle HTTP Server instances to the Business Process Management Managed Servers:

1. While the `WLS_SOA2` Managed Server is running, stop the `WLS_SOA1` Managed Server by using the Oracle WebLogic Remote Console.
2. Use your web browser to access the following URLs:

https://soa.example.com/bpm/composer/
https://soa.example.com/bpm/workspace/

3. Log in by using the `weblogic_soa` administration credentials.
You should see the BPM Composer and BPM Workspace applications (Figure 14-1 and Figure 14-2).
4. Start `WLS_SOA1` from the Oracle WebLogic Remote Console.
5. Stop `WLS_SOA2` from the Oracle WebLogic Remote Console.
6. Access the same URLs to verify that the load balancer and Oracle HTTP Server instances can route the requests to the other Managed Server.

Figure 14-1 Oracle BPM Composer

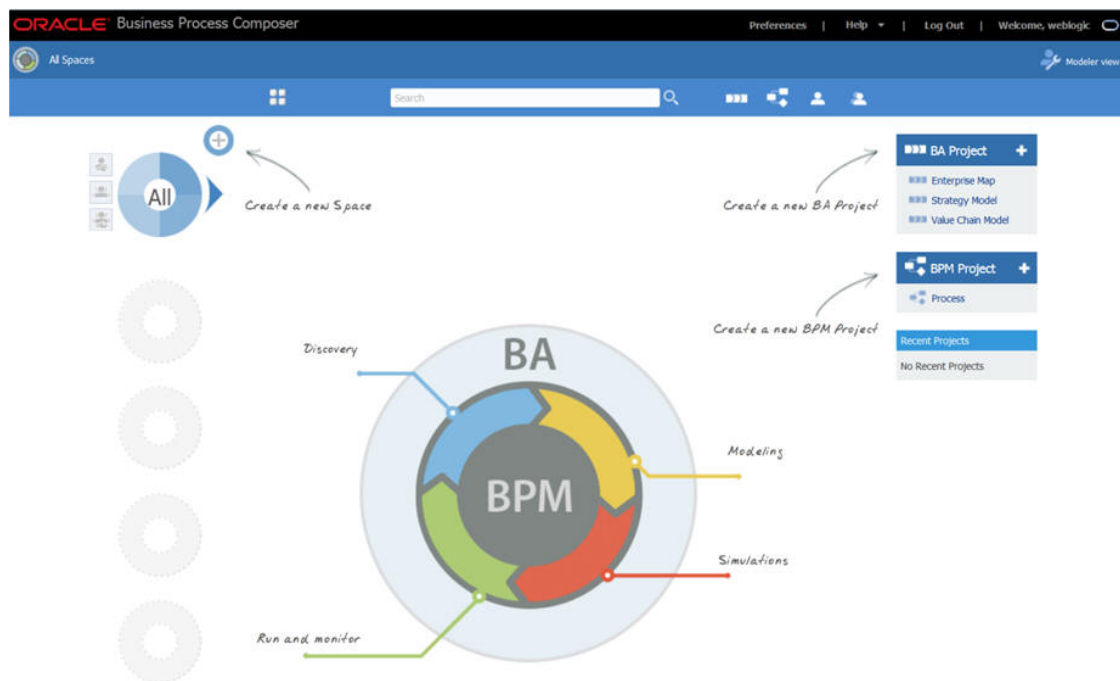
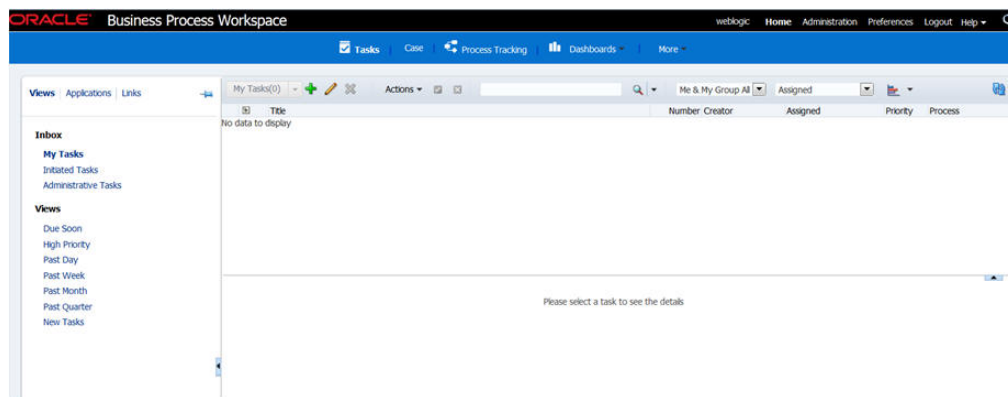


Figure 14-2 Oracle BPM Workspace



Configuring BPMJMSModule for the Oracle BPM Cluster

When you configure Oracle Business Process Management in a Oracle WebLogic Server domain, the BPMJMSModule JMS module is deployed automatically.

When you deploy Oracle Business Process Management Server as part of a Oracle WebLogic Server cluster, you must verify that the default values for the quota and redelivery limits for specific JMS resources within the BPMJMSModule JMS module are correct.

Specifically, you must verify the JMS topic resources listed in the following table.

Table 14-3 List of the JMS Topic Resources Within the BPMJMSModule JMS Module

JMS Resource	Property	Description	Recommended Setting
Measurement distributed topic in a cluster configuration: dist_MeasurementTopic_automato	Quota	If a large number of messages are published to the measurement JMS topic and the message consumption is relatively slow, this setting causes issues. When the JMS default threshold of maximum message size is reached, then additional messages cannot be published and any attempt at publishing fails with the following exception: ResourceAllocationException	Set Quota to MeasurementQuota
Measurement distributed topic in a cluster configuration: dist_MeasurementTopic_automato	Redelivery Limit	When this property is set to -1, JMS retries sending the message until the message is successfully acknowledged. If the measurement topic consumers cannot process messages due to a system error that causes the transaction to rollback, then the system can experience performance issues and the filling up of logs with repeated exceptions.	Set the redelivery limit to three (3).

Table 14-3 (Cont.) List of the JMS Topic Resources Within the BPMJMSModule JMS Module

JMS Resource	Property	Description	Recommended Setting
Measurement distributed topic in a cluster configuration: dist_MeasurementTopic_automato	Forwarding Policy	<p>A Forwarding Policy set to Replicated is not the best performance option for BPM analytics, verify that it is set to Partitioned.</p> <p>For more information about optimizing performance, see <i>Tuning Oracle Business Process Management in Tuning Performance</i>.</p> <p>For more information on partitioned and replicated forwarding policies, see <i>Configuring Partitioned Distributed Topics in Administering JMS Resources for Oracle WebLogic Server</i>.</p>	Set the Forwarding Policy to Partitioned .

To modify the BPMJMSModule resource settings:

1. Log into the WebLogic Remote Console.
2. In **Edit Tree**, navigate to **Services > JMS Modules**.
3. Click **BPMJMSModule**.
4. Expand **Uniform Distributed Topics** and click **dist_MeasurementTopic_auto_1**.
5. Click the **Thresholds** tab.
6. Verify that **Quota** is set to **MeasurementQuota**. If it is not set, select **MeasurementQuota** from the **Quota** drop-down menu and click **Save**.
7. Click the **Delivery Failure** tab.
8. Verify that the following fields are set to 3:
 - **Redelivery Delay Override**
 - **Redelivery Limit**
9. Click the **General** tab.
10. Verify that **Forwarding Policy** is set to **Partitioned**. If the default value is not **Partitioned**, select it and click **Save**.
11. Commit Changes.
12. Restart all SOA BPM cluster nodes for the changes to take effect.

Replacing Connect Strings with the Appropriate TNS Alias

Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

For more information about how to use TNS alias in your Datasources, see [Using TNS Alias in Connect Strings](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see [Performing Backups and Recoveries for an Enterprise Deployment](#).

Extending the Domain with Oracle Enterprise Scheduler

The procedures explained in this chapter guide you through the process of extending the enterprise deployment domain with the Oracle Enterprise Scheduler software.

- [About Adding Oracle Enterprise Scheduler](#)
Before you add Oracle Enterprise Scheduler to a SOA domain, familiarize yourself with the high-level steps that you have to perform to complete the extension process.
- [Variables Used When Configuring Oracle Enterprise Scheduler](#)
As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.
- [Support for Reference Configuration in Oracle Enterprise Scheduler](#)
Oracle Enterprise Scheduler supports both SOA Classic domains and SOA Reference Configuration domains. ESS can be added to a SOA Classic domain (created with the SOA Classic template) and to a SOA Reference Configuration domain (created with the SOA Reference Configuration template).
- [Creating the Database Schemas for ESS](#)
Before you can configure an Oracle ESS server, you must install the required schemas on a certified database for use with this release of Oracle Fusion Middleware.
- [Extending the SOA Domain to Include Oracle Enterprise Scheduler](#)
You can use the Configuration Wizard to configure and extend the existing enterprise deployment SOA domain with Oracle Enterprise Scheduler. You also need to perform additional tasks to complete the extension.
- [Update Certificates for New Frontend Addresses](#)
This section contains information about certificates for new frontend addresses.
- [Update the WebLogic Servers Security Settings](#)
This section contains information about WebLogic Servers security settings.
- [Propagating the Extended Domain to the Domain Directories and Machines](#)
After you have extended the domain with the ESS instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and machines.
- [Adding the ESSAdmin Role to the SOA Administrators Group](#)
Before you validate the Oracle Enterprise Scheduler configuration on the WLS_ESS1 Managed Server, add the `ESSAdmin` role to the enterprise deployment administration group (SOA Administrators).
- [Starting and Validating the WLS_ESS1 Managed Server](#)
Now that you have extended the domain, restarted the Administration Server, and propagated the domain to the other hosts, you can start the newly configured ESS servers.
- [Starting and Validating the WLS_ESS2 Managed Server](#)
After you start the WLS_ESS2 managed server, you must verify that the server status is reported as *Running* in the Remote Console and access the URLs to verify the status of servers.
- [Modifying the Upload and Stage Directories to an Absolute Path](#)

- [Configuring the Web Tier for the Extended Domain](#)
Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.
- [Validating Access to Oracle Enterprise Scheduler Through the Hardware Load Balancer](#)
Verify the URLs to ensure that appropriate routing and failover is working from the HTTP Server to the Oracle ESS components.
- [Replacing Connect Strings with the Appropriate TNS Alias](#)
Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.
- [Backing Up the Configuration](#)
It is an Oracle best practices recommendation to create a backup after you successfully extend 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.

About Adding Oracle Enterprise Scheduler

Before you add Oracle Enterprise Scheduler to a SOA domain, familiarize yourself with the high-level steps that you have to perform to complete the extension process.

[Table 15-1](#) lists and describes the high-level steps to extend a SOA domain with Oracle Enterprise Scheduler.

Table 15-1 Steps for Extending a SOA Domain to Include Oracle Enterprise Scheduler

Step	Description	More Information
Create Database Schemas for ESS	Navigate the RCU screens to create the database schemas.	Creating the Database Schemas for ESS
Run the Configuration Wizard to Extend the Domain	Extend the SOA/OSB domain to contain Oracle Enterprise Scheduler components.	Extending the SOA Domain to Include Oracle Enterprise Scheduler
Update Certificates for New Frontend Addresses	Since the ESS, SOA, and WSM servers use the same listen addresses (different port), there is no need to create new certificates and update stores.	Update Certificates for New Frontend Addresses
Update the WebLogic Servers Security Settings	Update SSL settings for the WLS_ESS1 and WLS_ESS2 servers.	Updating the WebLogic Servers Security Settings
Propagate the Domain Configuration to the Managed Server Directory in SOAHOST1 and to SOAHOST2	Oracle Enterprise Scheduler requires some updates to the WebLogic Server start scripts. Propagate these changes by using the <code>pack</code> and <code>unpack</code> commands.	Propagating the Extended Domain to the Domain Directories and Machines
Adding the ESSAdmin Role to the SOA Administrators Group	Before you validate the Oracle Enterprise Scheduler configuration on the WLS_ESS1 Managed Server, add the <code>ESSAdmin</code> role to the <code>enterprise deployment administration</code> group (SOA Administrators).	Adding the ESSAdmin Role to the SOA Administrators Group

Table 15-1 (Cont.) Steps for Extending a SOA Domain to Include Oracle Enterprise Scheduler

Step	Description	More Information
Start and validate the Oracle Enterprise Scheduler Servers	Once you have extended the domain, restarted the Administration Server, and propagated the domain to the other hosts, you can start the newly configured ESS servers. Verify that the server status is reported as Running in the Remote Console and access URLs to verify status of servers.	Starting and Validating the WLS_ESS1 Managed Server Starting and Validating the WLS_ESS2 Managed Server
Modifying the Upload and Stage Directories to an Absolute Path	After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters.	Modifying the Upload and Stage Directories to an Absolute Path
Configuring Oracle HTTP Server for the WLS_ESSn Managed Servers	To enable Oracle HTTP Server to route to Oracle Enterprise Scheduler console and service, set the WebLogicCluster parameter to the list of nodes in the cluster.	Configuring Oracle HTTP Server for the WLS_ESS Managed Servers
Validating Access Through Oracle HTTP Server	Verify that the server status is reported as Running.	Validating Access to Oracle Enterprise Scheduler Through the Hardware Load Balancer
Replacing Connect Strings with the Appropriate TNS Alias	Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.	Replacing Connect Strings with the Appropriate TNS Alias
Backing up the Oracle Enterprise Scheduler	To back up the domain configuration for immediate restoration in case of failures in future procedures.	Backing Up the Oracle Enterprise Scheduler Configuration

Variables Used When Configuring Oracle Enterprise Scheduler

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

The values for several directory variables are defined in [File System and Directory Variables Used in This Guide](#).

- *ORACLE_HOME*
- *ASERVER_HOME*
- *MSERVER_HOME*
- *WEB_DOMAIN_HOME*

In addition, you reference the following virtual IP (VIP) addresses that are defined in [Physical and Virtual IP Addresses Required by the Enterprise Topology](#):

- ADMINVHN

Actions in this chapter are performed on the following host computers:

- SOAHOST1

- SOAHOST2
- WEBHOST1
- WEBHOST2

Support for Reference Configuration in Oracle Enterprise Scheduler

Oracle Enterprise Scheduler supports both SOA Classic domains and SOA Reference Configuration domains. ESS can be added to a SOA Classic domain (created with the SOA Classic template) and to a SOA Reference Configuration domain (created with the SOA Reference Configuration template).

The process to extend a SOA Classic or a Reference Configuration domain to add ESS is the same and is described in this chapter.

Creating the Database Schemas for ESS

Before you can configure an Oracle ESS server, you must install the required schemas on a certified database for use with this release of Oracle Fusion Middleware.

Follow the instructions in these sections to install the schemas.

- [Starting the Repository Creation Utility \(RCU\)](#)
- [Navigating the RCU Screens to Create the Enterprise Scheduler Schemas](#)
- [Verifying Schema Access](#)

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Navigate to the `$ORACLE_HOME/oracle_common/bin` directory on your system.

```
cd $ORACLE_HOME/oracle_common/bin
```

2. Make sure that the `JAVA_HOME` environment variable is set to the location of a certified JDK on your system. The location should be up to but not including the `bin` directory. For example, if your JDK is located in `/u01/oracle/products/jdk`:

On UNIX operating systems:

```
export JAVA_HOME=/u01/oracle/products/jdk
```

3. Start RCU:

On UNIX operating systems:

```
./rcu
```

 **Note:**

If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces that are created by the RCU, provide the `-encryptTablespace true` option when you start RCU.

This defaults 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 Enterprise Scheduler Schemas

Schema creation involves the following tasks:

- [Task 1, Introducing RCU](#)
- [Task 2, Selecting a Method of Schema Creation](#)
- [Task 3, Providing Database Connection Details](#)
- [Task 4, Specifying a Custom Prefix and Selecting Schemas](#)
- [Task 5, Specifying Schema Passwords](#)
- [Task 6, Verifying the Tablespaces for the Required Schemas](#)
- [Task 7, Creating Schemas](#)
- [Task 8, Reviewing Completion Summary and Completing RCU Execution](#)

Task 1 Introducing RCU

Click **Next**.

Task 2 Selecting a Method of Schema Creation

If you have the necessary permission and privileges to perform DBA activities on your database, select **Create Repository > System Load and Product Load**. This procedure assumes that you have the necessary privileges.

If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option generates a SQL script, which can be provided to your database administrator. See Understanding System Load and Product Load in *Creating Schemas with the Repository Creation Utility*.

Click **Next**.

Task 3 Providing Database Connection Details

Provide the database connection details for RCU to connect to your database.

1. As Database Type, select **Oracle Database enabled for edition-based redefinition**.

 **Note:**

Oracle Database enabled for edition-based redefinition (EBR) is recommended to support Zero Down Time upgrades. For more information, see <https://www.oracle.com/database/technologies/high-availability/ebr.html>.

2. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
3. Enter the **Port** number of the RAC database scan listener, for example 1521.
4. Enter the RAC **Service Name** of the database.
5. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example `SYS`.
6. Enter the **Password** of the user name that you provided in step 4.
7. If you have selected the `SYS` user, ensure that you set the role to `SYSDBA`.
8. Click **Next** to proceed, then click **OK** on the dialog window confirming that connection to the database was successful.

Task 4 Specifying a Custom Prefix and Selecting Schemas

Select **Select existing prefix** and specify the prefix you used for the original domain creation schemas.

Expand the **Oracle AS Common Schemas** and then select the **Oracle Enterprise Scheduler** in the component list.

The custom prefix is used to logically group these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

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*.

Tip:

You must make a note of the custom prefix you choose to enter here; you need this later on during the domain creation process.

Click **Next** to proceed, then click **OK** on the dialog window confirming that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords

Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database security requirements before you continue. RCU proceeds at this point even if you do not meet the password policies. Hence, perform this check outside RCU itself.

Tip:

You must make a note of the passwords you set on this screen; you need them later on during the domain creation process.

Click **Next**.

Task 6 Verifying the Tablespaces for the Required Schemas

Click **Next** in the Default and temporary tablespaces selection (accept defaults), and click in the confirmation Pop-up window that warns about tablespaces that are being created.

Task 7 Creating Schemas

Review the summary of the schemas to be loaded, and click **Create** to complete schema creation.

Note:

If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and recreate the schemas before you continue.

Task 8 Reviewing Completion Summary and Completing RCU Execution

When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

Note:

If the database is a pluggable database (PDB), the appropriate tns alias that points to the PDB must be used in the sqlplus command.

```
./sqlplus FMW1412_ESS/<ess_password>
```

```
SQL*Plus: Release 23.0.0.0.0 - for Oracle Cloud and Engineered Systems on Wed Sep 11  
14:20:00 2024 Version 23.5.0.24.07  
Copyright (c) 1982, 2024, Oracle. All rights reserved.
```

```
Connected to:  
Oracle Database 23ai EE Extreme Perf Release 23.0.0.0.0 - for Oracle Cloud and  
Engineered Systems  
Version 23.5.0.24.07
```

```
SQL>
```

Extending the SOA Domain to Include Oracle Enterprise Scheduler

You can use the Configuration Wizard to configure and extend the existing enterprise deployment SOA domain with Oracle Enterprise Scheduler. You also need to perform additional tasks to complete the extension.

Extending the domain involves the following tasks.

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens to Extend the Domain with Oracle Enterprise Scheduler](#)

Starting the Configuration Wizard

Note:

SSL store customizations were added to the `setUserOverridesLate.sh` in the domain creation chapter. Any customizations added to this file are preserved when a domain is extended and are carried over to remote servers when using the **pack** and **unpack** commands.

However, if you added any additional customizations to the `setDomainEnv.sh` script in the domain (such as custom libraries, JAVA command line options for starting the servers or environment variables), those will be overwritten by the configuration wizard when you extend the domain. Add all the startup parameters that apply to all servers in a domain to the `setUserOverridesLate.sh` file. This will preserve them across extensions.

To begin domain configuration:

1. Shut down the Administration Server to prevent any configuration locks, saves, or activations from occurring during the configuration of the domain.
2. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

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

Navigating the Configuration Wizard Screens to Extend the Domain with Oracle Enterprise Scheduler

In this step, you extend the domain created in [Extending the Domain with Oracle SOA Suite](#) to contain Oracle Enterprise Scheduler components.

The steps reflected in this section are very similar to the steps required to extend an Oracle Fusion Middleware Infrastructure domain directly, but some of the options, libraries, and components shown in the screens will vary.

Domain creation and configuration includes the following tasks:

- [Selecting the Domain Type and Domain Home Location](#)
- [Selecting the Configuration Template](#)
- [Specifying the Database Configuration Type](#)
- [Specifying JDBC Component Schema Information](#)
- [Providing the GridLink Oracle RAC Database Connection Details](#)
- [Testing the JDBC Connections](#)
- [Selecting Advanced Configuration](#)
- [Configuring Managed Servers](#)
- [Configuring a Cluster](#)
- [Assigning Server Templates](#)
- [Configuring Dynamic Servers](#)
- [Assigning Managed Servers to the Cluster](#)
- [Configuring Coherence Clusters](#)
- [Verifying the Existing Machines](#)
- [Assigning Servers to Machines](#)
- [Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Writing Down Your Domain Home and Administration Server URL](#)
- [Start the Administration Server](#)

Selecting the Domain Type and Domain Home Location

On the Configuration Type screen, select **Update an existing domain**.

In the **Domain Location** field, select the value of the `ASERVER_HOME` variable, which represents the complete path to the Administration Server domain home you created in [Creating the Initial Infrastructure Domain for an Enterprise Deployment](#).

For more information about the directory location variables, see [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*.

Selecting the Configuration Template

On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle Enterprise Scheduler Service Basic [oracle_common]**
- **Oracle Enterprise Manager Plugin for ESS [em]**

Click **Next**.

Specifying the Database Configuration Type

On the Database Configuration Type screen, select **RCU Data**.

All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.

- Verify that **Vendor** is Oracle and **Driver** is *Oracle's Driver (Thin) for Service Connections; Versions: Any.
- Verify that **Connection Parameters** is selected.
- Verify and ensure that credentials in all the fields are the same as those provided during the configuration of Oracle Fusion Middleware Infrastructure.



Note:

Any custom datasources that were created before the extension (such as LEASING datasources) will show up before this screen. Check the Datasources row and click **Next**. The test datasource screen will verify its validity. Click **Next**.

Click **Get RCU Configuration** after you finish verifying 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.
```



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*.

Specifying JDBC Component Schema Information

Select the **ESS Schema** and **ESS MDS Schema**.

When you select the schemas, the fields on the page are activated and the database connection fields are populated automatically.

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

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 the following table.

Element	Description and Recommended Value
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, <i>soaedg.example.com</i> .
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)

Element	Description and Recommended Value
ONS Host and Port	<p>These values are not required when you are using an Oracle 12c database or higher versions because the ONS list is automatically provided from the database to the driver. Complete these values only if you are using Oracle 11g database:</p> <ul style="list-style-type: none"> In the ONS Host field, enter the SCAN address for the Oracle RAC database. In the Port field, enter the ONS Remote port (typically, 6200).
Enable Fan	Verify that the Enable Fan check box is selected, so the database can receive and process FAN events.

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.

For more information about the other options on this screen, see Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

Selecting Advanced Configuration

To complete domain configuration for the topology, select **Topology** on the Advanced Configuration screen.

Configuring Managed Servers

On the Managed Servers screen, add the required managed servers for Enterprise Scheduler.

- Select the automatically created server and click **Rename** to change the name to WLS_ESS1.
- Click **Add** to add another new server and enter WLS_ESS2 as the server name.
- Give servers WLS_ESS1 and WLS_ESS2 the attributes listed in [Table 15-2](#).

Click **Next**.

Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Server Groups	Administration Port
WLS_SOA1	SOAHOST1	Unchecked	Disabled	Checked	7004	SOA-MGD-SVRS-ONLY	9004
WLS_SOA2	SOAHOST2	Unchecked	Disabled	Checked	7004	SOA-MGD-SVRS-ONLY	9004
WLS_WSM1	SOAHOST1	Unchecked	Disabled	Checked	7010	JRF-MAN-SVR WSMPM-MAN-SVR	9003
WLS_WSM2	SOAHOST2	Unchecked	Disabled	Checked	7010	JRF-MAN-SVR WSMPM-MAN-SVR	9003
WLS_OSB1	SOAHOST1	Unchecked	Disabled	Checked	8003	OSB-MGD-SVRS-ONLY	9007

Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Server Groups	Administrati on Port
WLS_OSB2	SOAHOST2	Unchecked	Disabled	Checked	8003	OSB-MGD-SVRS-ONLY	9007
WLS_ESS1	SOAHOST1	Unchecked	Disabled	Checked	7008	ESS-MGD-SVRS	9006
WLS_ESS2	SOAHOST2	Unchecked	Disabled	Checked	7008	ESS-MGD-SVRS	9006

 **Note:**

- The WLS_SOA Managed Servers appear only if you are extending a domain where Oracle SOA Suite has been configured.
- The WLS_OSB Managed Servers appear only if you are extending a domain where Oracle Service Bus has been configured.

Configuring a Cluster

On the Configure Clusters screen, add the **ESS_Cluster** cluster, by using the values for each cluster as shown in [Table 19-1 The Frontend Hostname and Port for Each Cluster](#) in [Setting the Front End Host and Port for a WebLogic Cluster](#).

Click **Next**.

Assigning Server Templates

Click **Next**.

Configuring Dynamic Servers

Click **Next**.

Assigning Managed Servers to the Cluster

On the Assign Servers to Clusters screen, assign servers to clusters as follows:

- SOA_Cluster - If you are extending a SOA domain.
 - WLS_SOA1
 - WLS_SOA2
- WSM-PM_Cluster:
 - WLS_WSM1
 - WLS_WSM2
- OSB_Cluster - If you are extending an OSB domain:
 - WLS_OSB1
 - WLS_OSB2
- ESS_Cluster:
 - WLS_ESS1
 - WLS_ESS2

Click **Next**.

Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at 9991, as it was defined during the initial Infrastructure domain creation.

Verifying the Existing Machines

On the Unix Machines tab, confirm that the following entries appear:

Name	Node Manager Listen Address
SOAHOST1	SOAHOST1
SOAHOST2	SOAHOST2
ADMINHOST	ADMINVHN

Leave all other fields to their default values.
Click **Next**.

Assigning Servers to Machines

On the Assign Servers to Machines screen, assign servers to machines as follows:

- ADMINHOST:
 - AdminServer
- SOAHOST1
 - WLS_SOA1 (if extending a SOA domain)
 - WLS_WSM1
 - WLS_OSB1 (if extending an OSB domain)
 - WLS_ESS1
- SOAHOST2
 - WLS_SOA2 (if extending a SOA domain)
 - WLS_WSM2
 - WLS_OSB2 (if extending an OSB domain)
 - WLS_ESS2

Click **Next**.

Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the detailed configuration information for the domain you are about to extend. Review the details of each item on the screen and verify that the information is correct.

If you need to make any changes, you can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.

Click **Update** to execute the domain extension.

In the Configuration Progress screen, click **Next** when it finishes.

For more information about the options on this screen, see Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen will show the following items about the domain you just configured, including:

- Domain Location
- Administration Server URL

Make a note of both these items, because you will need them later; you will need the domain location to access the scripts used to start the Administration Server, and you will need the Administration Server URL to access the WebLogic Remote Console and Oracle Enterprise Manager Fusion Middleware Control.

Click **Finish** to dismiss the Configuration Wizard.

If the Admin Server was running during the domain extension process, restart the server before you continue.

Start the Administration Server

Start the Administration Server to ensure the changes you have made to the domain have been applied.

After you have completed extending the domain, go to [Propagating the Extended Domain to the Domain Directories and Machines](#).

Update Certificates for New Frontend Addresses

This section contains information about certificates for new frontend addresses.

Note:

About Certificates for the domain extension.

Since the ESS, SOA, and WSM servers use the same listen addresses (different port), there is no need to create new certificates and update stores. Also, since the ESS cluster uses the same frontend address as the SOA cluster, there is no need to update the OHS certificates and update trusted keystores. No action is required.

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in the [Updating the WebLogic Servers Security Settings](#) and update SSL settings for the WLS_ESS1 and WLS_ESS2 servers.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the ESS instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and machines.

The following table summarizes the steps that are required to propagate the changes to all the domain directories and machines.

Task	Description	More Information
Pack up the Extended Domain on SOAHOST1	Use the <code>pack</code> command to create a new template jar file that contains the new ESS Servers configuration. When you pack up the domain, create a template jar file called <code>soadomaintemplateExtESS.jar</code> .	Packing Up the Extended Domain on SOAHOST1
Unpack the Domain in the Managed Servers Directory on SOAHOST1	Unpack the template jar file in the Managed Servers directory on SOAHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1
Unpack the Domain on SOAHOST2	Unpack the template jar file in the Managed Servers directory on the SOAHOST2 local storage.	Unpacking the Domain on SOAHOST2

Adding the ESSAdmin Role to the SOA Administrators Group

Before you validate the Oracle Enterprise Scheduler configuration on the WLS_ESS1 Managed Server, add the `ESSAdmin` role to the enterprise deployment administration group (SOA Administrators).

To perform this task, refer to [Configuring Roles for Administration of Oracle SOA Suite Products](#).

Starting and Validating the WLS_ESS1 Managed Server

Now that you have extended the domain, restarted the Administration Server, and propagated the domain to the other hosts, you can start the newly configured ESS servers.

To start the configured ESS servers:

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

```
https://ADMINVHN:9002/em
```

In this example:

- Replace ADMINVHN with the host name assigned to the ADMINVHN Virtual IP address in [Identifying and Obtaining Software Downloads for an Enterprise Deployment](#).
 - Port 9002 is the typical Administration port used for the WebLogic Remote Console and Fusion Middleware Control.
2. Log in to Fusion Middleware Control by using the Administration Server credentials.
 3. In the Target Navigation pane, expand the domain to view the Managed Servers in the domain.
 4. Select only the WLS_ESS1 Managed Server, and click **Start Up** on the Oracle WebLogic Server tool bar.

 **Note:**

ESS Servers depend on the policy access service to be functional. This implies that the WSM-PM servers in the domain need to be reachable before the SOA servers are started.

5. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_ESS1 Managed Server is up and running.
6. To verify the ESS software is configured, enter the following URL in the browser:

`https://SOAHOST1:7008/EssHealthCheck/`

With the default installation, this should be the HTTP response, as shown in the following image.

ESS - Diagnostic health check service



Click on the **Check Health** button, and then log in by using the `welogic_soa` administration credentials.

The reply should report that Oracle Enterprise Schedule (ESS) is up and running.

Starting and Validating the WLS_ESS2 Managed Server

After you start the WLS_ESS2 managed server, you must verify that the server status is reported as *Running* in the Remote Console and access the URLs to verify the status of servers.

Perform the same steps that you used to start WLS_ESS1, to start WLS_ESS2.

1. Log in to Fusion Middleware Control by using the Administration Server credentials.
2. In the Target Navigation pane, expand the domain to view the Managed Servers in the domain.
3. Select only the WLS_ESS2 Managed Server, and click **Start Up** on the Oracle WebLogic Server tool bar.
4. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_ESS2 Managed Server is up and running, access the equivalent URLs for the WLS_ESS2:

`https://SOAHOST2:7008/EssHealthCheck/`

Click the **Check Health** button, and then log in by using the `welogic_soa` administration credentials.

The reply reports that Oracle Enterprise Scheduler is up and running.

Modifying the Upload and Stage Directories to an Absolute Path

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. See [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).

Configuring the Web Tier for the Extended Domain

Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.

- [Configuring Oracle HTTP Server for the WLS_ESS Managed Servers](#)
Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle Enterprise Scheduler requests correctly to the WLS_ESS Managed Servers on SOHOST1 and SOAHOST2.

Configuring Oracle HTTP Server for the WLS_ESS Managed Servers

Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle Enterprise Scheduler requests correctly to the WLS_ESS Managed Servers on SOHOST1 and SOAHOST2.

To enable Oracle HTTP Server to route Oracle Enterprise Scheduler requests to the application tier:

1. Log in to SOAHOST1 and change directory to the configuration directory for the first Oracle HTTP Server instance (ohs1):

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

2. Add the following directives inside the `<VirtualHost>` tag in the `soa_vh.conf` file:

Note:

The `WebLogicCluster` directive needs only a sufficient number of redundant `server:port` combinations to guarantee initial contact in case of a partial outage. The actual total list of cluster members is retrieved automatically upon first contact with any given node.

```
<Location /ess >
  WLSRequest ON
  WebLogicCluster SOAHOST1:7008,SOAHOST2:7008
</Location>

<Location /EssHealthCheck >
  WLSRequest ON
  WebLogicCluster SOAHOST1:7008,SOAHOST2:7008
</Location>
```

```

<Location /ess-async >
  WLSRequest ON
  WebLogicCluster SOAHOST1:7008,SOAHOST2:7008
</Location>

<Location /ess-wsjob >
  WLSRequest ON
  WebLogicCluster SOAHOST1:7008,SOAHOST2:7008
</Location>

```

3. Log in to SOAHOST2 and change directory to the following location so you can update the configuration file for the second Oracle HTTP Server instance (ohs2):

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf
```

4. Open the `soa_vh.conf` file and add the Oracle Enterprise Scheduler directives to the `<VirtualHost>` tag.
5. Restart Oracle HTTP Servers on WEBHOST1 and WEBHOST2.

Validating Access to Oracle Enterprise Scheduler Through the Hardware Load Balancer

Verify the URLs to ensure that appropriate routing and failover is working from the HTTP Server to the Oracle ESS components.

To verify the URLs:

1. While WLS_ESS1 is running, stop WLS_ESS2 by using the Oracle WebLogic Remote Console.
2. Access the following URL from your web browser, and verify the HTTP response as indicated in [Starting and Validating the WLS_ESS2 Managed Server](#):

```
https://soa.example.com/EssHealthCheck
```

3. Start WLS_ESS2 from the Oracle WebLogic Remote console.
4. Stop WLS_ESS1 from the Oracle WebLogic Remote console.
5. Verify these URLs by using your load balancer address:

```
https://soa.example.com/EssHealthCheck
https://soa.example.com/ess
```

Replacing Connect Strings with the Appropriate TNS Alias

Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

For more information about how to use TNS alias in your Datasources, see [Using TNS Alias in Connect Strings](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extend a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see [Performing Backups and Recoveries in the SOA Enterprise Deployments](#).

16

Extending the Domain with Business Activity Monitoring

The procedures explained in this chapter guide you through the process of extending the enterprise deployment domain to include Oracle Business Activity Monitoring.

- [Variables Used When Configuring Business Activity Monitor](#)
As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.
- [Support for Reference Configuration in BAM](#)
Oracle BAM does not support Reference Configuration. Hence, you can add BAM only to a Classic SOA domain.
- [About Configuring BAM in Its Own Domain](#)
For adding BAM to the enterprise topology, you can add it to the existing SOA domain or you can create a new domain for BAM, separate from the Oracle SOA suite domain.
- [Prerequisites When Adding Oracle BAM to the Domain](#)
Before you add Oracle BAM to your existing Oracle SOA Suite domain, consider the following information and prerequisites.
- [Special Instructions When Configuring Oracle BAM on Separate Hosts](#)
If you choose to configure Oracle BAM on its own hardware, then you can use the instructions in this chapter, as long as you also consider the information in the following sections.
- [Roadmap for Adding Oracle BAM to the Domain](#)
The table in this section lists the high-level steps to extend a SOA domain for Oracle Business Activity Monitoring.
- [Extending the SOA Domain to Include Oracle Business Activity Monitoring](#)
You can use the Configuration Wizard to extend the existing enterprise deployment SOA domain with the Oracle Business Activity Monitoring.
- [Update Certificates for New Frontend Addresses](#)
This section contains information about certificates for new frontend addresses.
- [Update the WebLogic Servers Security Settings](#)
This section contains information about WebLogic Servers security settings.
- [Propagating the Extended Domain to the Domain Directories and Machines](#)
After you have extended the domain with the BAM instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and machines.
- [Adding the Enterprise Deployment Administration User to the Oracle BAM Administration Group](#)
Before you validate the Oracle BAM configuration on the Managed Server, add the enterprise deployment administration user (weblogic_soa) to the BAMAdministrator group.
- [Starting and Validating the WLS_BAM1 Managed Server](#)
After extending the domain, restarting the Administration Server, and propagating the domain to the other hosts, start the newly configured BAM servers.

- [Starting and Validating the WLS_BAM2 Managed Server](#)
After you start the WLS_BAM2 managed server, you must verify that the server status is reported as *Running* in the Remote Console and access the URLs to verify the status of the servers.
- [Modifying the Upload and Stage Directories to an Absolute Path](#)
- [Configuring the Web Tier for the Extended Domain](#)
Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.
- [Validating Access to Oracle BAM Through the Hardware Load Balancer](#)
Verify that Oracle BAM URLs are successfully routing requests from the hardware load balancer to the Oracle HTTP Server instances to the Oracle BAM software in the middle tier.
- [Replacing Connect Strings with the Appropriate TNS Alias](#)
Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.
- [Backing Up the Configuration](#)
It is an Oracle best practices recommendation to create a backup after you successfully extend 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 Business Activity Monitor

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

The values for several directory variables are defined in [File System and Directory Variables Used in This Guide](#).

- `ORACLE_HOME`
- `ASERVER_HOME`
- `MSERVER_HOME`
- `ORACLE_RUNTIME`
- `WEB_DOMAIN_HOME`

In addition, you reference the following virtual IP (VIP) address that are defined in [Physical and Virtual IP Addresses Required by the Enterprise Topology](#):

- `ADMINVHN`

Actions in this chapter are performed on the following host computers:

- `SOAHOST1`
- `SOAHOST2`
- `WEBHOST1`
- `WEBHOST2`
- `BAMHOST1`
- `BAMHOST2`

Support for Reference Configuration in BAM

Oracle BAM does not support Reference Configuration. Hence, you can add BAM only to a Classic SOA domain.

About Configuring BAM in Its Own Domain

For adding BAM to the enterprise topology, you can add it to the existing SOA domain or you can create a new domain for BAM, separate from the Oracle SOA suite domain.

For more information about building the SOA topology, see [Building Your Own Oracle SOA Suite Enterprise Topology](#).

If you decide to configure BAM in a separate domain, keep the following points in mind to add BAM to your topology:

- Ignore any references to the SOA Managed Servers or the SOA Cluster. These elements of the domain exist only if you extend a domain that has already been extended with the Oracle SOA suite.
- Run the Repository Creation Utility (RCU) to create the SOAINFRA schema for the BAM domain. This schema is required by BAM. You must use a unique SOAINFRA schema and schema prefix for the BAM domain.
- When running the Configuration Wizard, the *High Availability Options* screen appears as described in [Navigating the Configuration Wizard Screens to Extend the Domain with Oracle SOA Suite](#).

This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply HA options (that is, the Configuration Wizard creates the JDBC stores and configures ASM for them).

Oracle recommends that you select the following options to configure Automatic Service Migration and JDBC stores automatically:

- Select **Enable Automatic Service Migration** with **Database Basis**.
- Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
- Set **JMS Server Persistence** to **JMS JDBC Store**.

Prerequisites When Adding Oracle BAM to the Domain

Before you add Oracle BAM to your existing Oracle SOA Suite domain, consider the following information and prerequisites.

Note:

If you choose to install Oracle BAM on a separate set of host computers, then in addition to the prerequisites listed here, see [Special Instructions When Configuring Oracle BAM on Separate Hosts](#).

- [Understanding the Installation Requirements for Adding Oracle BAM to the Domain](#)

- [Understanding the Database Schema Requirements for Oracle BAM](#)
- [Backing Up the Existing Installation](#)

Understanding the Installation Requirements for Adding Oracle BAM to the Domain

This chapter assumes that you are configuring Oracle Business Activity Monitoring on the same host computers as Oracle SOA Suite, as shown in [Figure 3-2](#).

In the default Oracle SOA Suite and Oracle Business Activity Monitoring topology, you target Oracle BAM to its own Managed Servers and its own cluster, but it shares system resources with the other Oracle SOA Suite products on SOAHOST1 and SOAHOST2. Those system resources include a shared storage device where the Oracle SOA Suite software has been installed in an existing Oracle home directory.

In the default topology, there is no need to install Oracle BAM, because Oracle BAM is included in the Oracle SOA Suite and Oracle Business Process Management distribution and is installed into the Oracle home directories when you install Oracle SOA Suite in [Understanding the SOA Enterprise Deployment Topology](#).

Understanding the Database Schema Requirements for Oracle BAM

The schemas required for Oracle BAM are created in the database when you run the Repository Creation Utility (RCU) to create the required Oracle SOA Suite schemas.

As a result, there is no need to run RCU specifically for Oracle BAM.

Backing Up the Existing Installation

If you have not yet backed up the existing Fusion Middleware Home and domain, back it up now.

To back up the existing Fusion Middleware Home and domain, see [Performing Backups and Recoveries in the SOA Enterprise Deployments](#).

Special Instructions When Configuring Oracle BAM on Separate Hosts

If you choose to configure Oracle BAM on its own hardware, then you can use the instructions in this chapter, as long as you also consider the information in the following sections.

For some organizations, it might make sense to install and configure Oracle BAM on separate host computers so the Oracle BAM software can use dedicated hardware resources and can be further isolated from the other Oracle SOA Suite products.

- [Procuring Additional Host Computers for Oracle BAM](#)
- [Installation Requirements When Configuring Oracle BAM on Separate Hosts](#)
- [Configuration Wizard Instructions When Configuring Oracle BAM on Separate Hosts](#)
- [Propagating the Domain Configuration When Configuring Oracle BAM on Separate Hosts](#)

Procuring Additional Host Computers for Oracle BAM

If you are configuring Oracle BAM on its own set of host computers, you must procure the additional hardware and be sure that it meets the system requirements described in [Host Computer Hardware Requirements](#) and [Operating System Requirements for the Enterprise Deployment Topology](#).

You should also add the required entries to the Enterprise Deployment Workbook, as described in [Using the Enterprise Deployment Workbook](#). For the purposes of this guide, you can refer to these host computers as BAMHOST1 and BAMHOST2.

Installation Requirements When Configuring Oracle BAM on Separate Hosts

If you configure Oracle BAM on its own set of host computers, then you should follow the same shared storage strategy that you are following for the host computers where the other Oracle SOA Suite products are installed.



Note:

The Oracle home used by BAMHOST1 and BAMHOST2 must contain the exact set of software binaries used by the SOAHOST1 and SOAHOST2 hosts in the domain; otherwise, unpredictable behavior in the execution of the binaries may occur.

Depending on your shared storage strategy, one of the following sections apply if you are using separate host hardware for the Oracle BAM software:

- [Installation Requirements When Using a Separate Volume or Partition](#)
- [Installation Requirements When Using a Shared Oracle Home](#)

Installation Requirements When Using a Separate Volume or Partition

If BAMHOST1 and BAMHOST2 are using separate shared storage volumes or partitions, then you must install the Infrastructure and optionally Oracle SOA Suite on those hosts. For more information, see [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

Note that the location where you install the Oracle home (which contains the software binaries) varies, depending upon the host. To identify the proper location for your Oracle home directories, refer to the guidelines in [File System and Directory Variables Used in This Guide](#).

To install the software on BAMHOST1 and BAMHOST2, log in to each host, and perform the following tasks:

- Use the instructions in [Installing the Oracle Fusion Middleware Infrastructure in Preparation for an Enterprise Deployment](#) to create the Oracle home on the appropriate storage device and install Oracle Fusion Middleware Infrastructure.
- Optionally, use the instructions in [Installing Oracle SOA Suite for an Enterprise Deployment](#) to install the Oracle SOA Suite software.

Installation Requirements When Using a Shared Oracle Home

If BAMHOST1 and BAMHOST2 are using an existing volume or partition where the Oracle Fusion Middleware Infrastructure or Oracle SOA Suite are already installed, then you must mount the volumes appropriately to BAMHOST1 and BAMHOST2. For more information, see [Mounting the Required Shared File Systems on Each Host](#). Ensure that BAMHOST1 and BAMHOST2 have access to this Oracle home, similar to the rest of the hosts in the domain.

This is the preferred method of using shared storage for the enterprise deployment. For more information, see [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

After you have mounted an existing volume or partition that contains an existing Oracle home, then you should attach the Oracle home to the local Oracle Inventory on BAMHOST1 or BAMHOST2.

To attach an Oracle home in shared storage to the local Oracle Inventory, use the following command on the BAMHOSTs:

```
cd $ORACLE_HOME/oui/bin/  
./attachHome.sh -jreLoc JAVA_HOME
```

The `pack` and `unpack` utilities is used to bootstrap the domain configuration for the WLS_BAM1 and WLS_BAM2 servers. As a result, if you have mounted an existing Oracle home with the required software already installed, then you do not need to install any software in these two hosts.

Configuration Wizard Instructions When Configuring Oracle BAM on Separate Hosts

If you configure Oracle BAM on separate host computers, then the instructions in this chapter for configuring the domain with the Configuration Wizard are slightly different.

Specifically, be sure to create additional Oracle WebLogic Server machines for BAMHOST1 and BAMHOST2, and then target the WLS_BAM1 and WLS_BAM2 Managed Servers to those machines, rather than to SOAHOST1 and SOAHOST2. See [Task 14, Verifying the Existing Machines](#) and [Task 15, Assigning Servers to Machines](#).

Propagating the Domain Configuration When Configuring Oracle BAM on Separate Hosts

If you configure Oracle BAM on separate host computers, then the instructions in this chapter for propagating the domain to the other domain directories must be modified.

Specifically, in addition to propagating the domain to the Managed Server domain directories on SOAHOST1 and SOAHOST2, you must also unpack the domain in the local Managed Server directories for BAMHOST1 and BAMHOST2.

Note that this means you must start the Node Manager software on each BAMHOST computer before you can remotely start the WLS_BAM Managed Servers on these hosts.

Roadmap for Adding Oracle BAM to the Domain

The table in this section lists the high-level steps to extend a SOA domain for Oracle Business Activity Monitoring.

Step	Description	More Information
Run the Configuration Wizard to Extend the Domain in the Administration Server domain home	Extend the SOA domain to contain Oracle BAM components.	Extending the SOA Domain to Include Oracle Business Activity Monitoring
Update Certificates for New Frontend Addresses	Since the BAM, SOA, and WSM servers use the same listen addresses (different port), there is no need to create new certificates and update stores.	Update Certificates for New Frontend Addresses
Update the WebLogic Servers Security Settings	Update SSL settings for the WLS_BAM1 and WLS_BAM2 servers.	Updating the WebLogic Servers Security Settings
Propagate the Domain Configuration to the Managed Server domain directories	Oracle BAM requires some updates to the WebLogic Server start scripts. Propagate these changes by using the <code>pack</code> and <code>unpack</code> commands.	Propagating the Extended Domain to the Domain Directories and Machines
Add the SOA Administrator role to the Oracle BAM Administration Group	This step allows you to use one set of credentials to access the various product-specific management utilities.	Adding the Enterprise Deployment Administration User to the Oracle BAM Administration Group
Start and validate the Oracle BAM Servers	Once you have extended the domain, restarted the Administration Server, and propagated the domain to the other hosts, you can start the newly configured BAM servers. Verify that the server status is reported as Running in the Remote Console and access URLs to verify status of servers.	Starting and Validating the WLS_BAM1 Managed Server Starting and Validating the WLS_BAM2 Managed Server
Modifying the Upload and Stage Directories to an Absolute Path	After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters.	Modifying the Upload and Stage Directories to an Absolute Path
Configuring Oracle HTTP Server for the WLS_BAMn Managed Servers	To enable Oracle HTTP Server to route to Oracle BAM, add the required directives to the Oracle HTTP Server configuration files, and set the <code>WebLogicCluster</code> parameter to the list of nodes in the cluster.	Configuring Oracle HTTP Server for the WLS_BAM Managed Servers
Validating Access Through Oracle HTTP Server	Verify that the server status is reported as Running.	Validating Access to Oracle BAM Through the Hardware Load Balancer
Replacing Connect Strings with the Appropriate TNS Alias	Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.	Replacing Connect Strings with the Appropriate TNS Alias
Backing up the Oracle BAM Configuration	To back up the domain configuration for immediate restoration in case of failures in future procedures.	Backing Up the Oracle BAM Configuration

Extending the SOA Domain to Include Oracle Business Activity Monitoring

You can use the Configuration Wizard to extend the existing enterprise deployment SOA domain with the Oracle Business Activity Monitoring.

Extending the domain involves the following tasks.

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens for Oracle BAM](#)

Starting the Configuration Wizard

Note:

SSL store customizations were added to the `setUserOverridesLate.sh` in the domain creation chapter. Any customizations added to this file are preserved when a domain is extended and are carried over to remote servers when using the **pack** and **unpack** commands.

However, if you added any additional customizations to the `setDomainEnv.sh` script in the domain (such as custom libraries, JAVA command line options for starting the servers or environment variables), those will be overwritten by the configuration wizard when you extend the domain. Add all the startup parameters that apply to all servers in a domain to the `setUserOverridesLate.sh` file. This will preserve them across extensions.

To begin domain configuration:

1. Shut down the Administration Server to prevent any configuration locks, saves, or activations from occurring during the configuration of the domain.
2. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

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

Navigating the Configuration Wizard Screens for Oracle BAM

In this step, you extend the domain created in [Extending the Domain with Oracle SOA Suite](#), to contain Oracle Business Activity Monitoring components.

The steps reflected in this section would be very similar if Oracle Business Activity Monitoring was extending a domain containing only an Administration Server and a WSM-PM Cluster, but some of the options, libraries and components shown in the screens could vary.

Domain creation and configuration includes the following tasks:

- [Task 1, Selecting the Domain Type and Domain Home Location](#)
- [Task 2, Selecting the Configuration Template](#)
- [Task 3, Specifying the Database Configuration Type](#)

- [Task 4, Specifying JDBC Component Schema Information](#)
- [Task 5, Providing the GridLink Oracle RAC Database Connection Details](#)
- [Task 6, Testing the JDBC Connections](#)
- [Task 7, Selecting Advanced Configuration](#)
- [Task 8, Configuring Managed Servers](#)
- [Task 9, Configuring a Cluster](#)
- [Task 10, Assigning Server Templates](#)
- [Task 11, Configuring Dynamic Servers](#)
- [Task 12, Assigning Managed Servers to the Cluster](#)
- [Task 13, Configuring Coherence Clusters](#)
- [Task 14, Verifying the Existing Machines](#)
- [Task 15, Assigning Servers to Machines](#)
- [Task 16, Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Task 17, Writing Down Your Domain Home and Administration Server URL](#)
- [Task 18, Start the Administration Server](#)

Task 1 Selecting the Domain Type and Domain Home Location

On the Configuration Type screen, select **Update an existing domain**.

In the **Domain Location** field, select the value of the `ASERVER_HOME` variable, which represents the complete path to the Administration Server domain home that you created in [Creating the Initial Infrastructure Domain for an Enterprise Deployment](#).

For more information about the directory location variables, see [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 Template

On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following template:

- **Oracle Business Activity Monitoring [soa]**

Click **Next**.

Task 3 Specifying the Database Configuration Type

On the Database Configuration Type screen, select **RCU Data**.

All fields are pre-populated, because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain.

- Verify that the **Vendor** is `Oracle` and the **Driver** is `*Oracle's Driver (Thin) for Service Connections; Versions: Any`.
- Verify that **Connection Parameters** is selected.

- Verify and ensure that credentials in all the fields are the same as those provided during the configuration of Oracle Fusion Middleware Infrastructure.



Note:

Any custom data sources that were created before the extension (such as LEASING datasources) shows up before this screen. Check the Datasources row and click **Next**. The test data source screen verifies its validity. Click **Next**.

Click **Get RCU Configuration** after you finish verifying 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.
```

Task 4 Specifying JDBC Component Schema Information

On the JDBC Component Schema page, select the following schemas:

- **BAM Schema**
- **BAM Job Sched Schema**
- **BAM Leasing Schema**
- **BAM Non JTA Schema**
- **BAM MDS Schema**

Select **Convert to Gridlink**, and then click **Next**.

Task 5 Providing the GridLink Oracle RAC Database Connection Details

On the GridLink Oracle RAC Component Schema screen, provide the information that is required to connect to the RAC database and component schemas, as shown in the following table.

Element	Description and Recommended Value
Service Name	Verify that the service name for the Oracle RAC database is appropriate. For example, <code>soaedg.example.com</code> .
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	These values are not required when you are using an Oracle 12c database or higher versions because the ONS list is automatically provided from the database to the driver.
Enable Fan	Verify that the Enable Fan check box is selected, so the database can receive and process FAN events.

Task 6 Testing the JDBC Connections

On the Test JDBC Data Sources screen, confirm that all connections were successful. The connections are tested automatically. The Status column displays the results. If all connections are not successful, click **Previous** to return to the previous screen and correct your entries.

Click **Next** when all the connections are successful.

Task 7 Selecting Advanced Configuration

To complete domain configuration for the topology, select **Topology** on the Advanced Configuration screen.

**Note:**

JDBC stores are recommended and selected in [Task 3, Configuring High Availability Options](#) so there is no need to configure File Stores.

If you choose File Stores in [Task 3, Configuring High Availability Options](#), you have to select the File Stores option here to configure them in a shared location in `ORACLE_RUNTIME/domain_name/BAM_Cluster/jms`. Shared location is required to resume JMS and JTA in a failover scenario.

Click **Next**.

Task 8 Configuring Managed Servers

On the Managed Servers screen, add the required managed servers for Oracle BAM:

- Select the automatically created server and rename it to WLS_BAM1.
- Click **Add** to add another new server and enter WLS_BAM2 as the server name.
- Select **BAM12-MGD-SVRS-ONLY** as the server group for the BAM Servers. Deselect **BAM12-MGD-SVRS** from the list.

The configuration for the added servers should match those shown in the following table.

Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Administrati on Port	Server Groups
WLS_SOA1*	SOAHOST1	Unchecked	Disabled	Checked	7004	9004	SOA-MGD-SVRS-ONLY
WLS_SOA2*	SOAHOST2	Unchecked	Disabled	Checked	7004	9004	SOA-MGD-SVRS-ONLY
WLS_WSM1	SOAHOST1	Unchecked	Disabled	Checked	7010	9003	JRF-MAN-SVR WSMPM-MAN-SVR
WLS_WSM2	SOAHOST2	Unchecked	Disabled	Checked	7010	9003	JRF-MAN-SVR WSMPM-MAN-SVR
WLS_BAM1	SOAHOST1	Unchecked	Disabled	Checked	7006	9005	BAM12-MGD-SVRS-ONLY
WLS_BAM2	SOAHOST2	Unchecked	Disabled	Checked	7006	9005	BAM12-MGD-SVRS-ONLY

*The WLS_SOA1 and WLS_SOA2 Managed Servers are shown if you extend a domain where Oracle SOA Suite has already been configured.

*When you specify the listen address for WLS_BAM1 and WLS_BAM2, enter SOAHOST1 and SOAHOST2, respectively, unless you configure Oracle BAM on separate host computers (BAMHOST1 and BAMHOST2). If you configure Oracle BAM on separate hosts, enter BAMHOST1 and BAMHOST2.

Task 9 Configuring a Cluster

On the Configure Clusters screen, click **Add** to add the **BAM_Cluster** (leave the present cluster as they are):

Name	Cluster Address	Frontend Host	Frontend HTTP Port	Frontend HTTPS Port
SOA_Cluster*	Leave it empty	soa.example.com	0	443
WSM-PM_Cluster	Leave it empty	internal.example.com	0	444
BAM_Cluster	Leave it empty	soa.example.com	0	443

*The SOA cluster appears only if you have already configured Oracle SOA Suite in the domain.

Click **Next**.

Task 10 Assigning Server Templates

Click **Next**.

Task 11 Configuring Dynamic Servers

Click **Next**.

Task 12 Assigning Managed Servers to the Cluster

On the Assign Servers to Clusters screen, assign servers to clusters as follows:

- BAM_Cluster:
 - WLS_BAM1
 - WLS_BAM2

Click **Next**.

Task 13 Configuring Coherence Clusters

Use the Coherence Clusters screen to configure the Coherence cluster that is automatically added to the domain. Leave the port number value at 9991, as it was defined during the initial Infrastructure domain creation.

Task 14 Verifying the Existing Machines

Verify the machines that have already been created in the domain. By default, you are targeting the new Oracle BAM Managed Servers to the SOAHOST1 and SOAHOST2 machines, respectively.

However, if you configure Oracle BAM on separate host computers, then you must create two new machines for the corresponding BAMHOST1 and BAMHOST2 host computers:

1. Select the **Unix Machine** tab.
2. Use the **Add** button to create two new Unix machines for BAMHOST1 and BAMHOST2.

Node Manager Listen Address to the physical IP address for BAMHOST1 and BAMHOST2.

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.

Leave all other fields to their default values.

Click **Next**.

Task 15 Assigning Servers to Machines

On the Assign Servers to Machines screen, assign the new WLS_BAM1 and WLS_BAM2 servers to the SOAHOST1 and SOAHOST2 machines, respectively.

However, if you are configuring Oracle BAM on separate host computers, assign the new Oracle BAM servers to the newly created BAMHOST1 and BAMHOST2 machines, respectively.

Click **Next**.

Task 16 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the detailed configuration information for the domain you are about to extend. Review the details of each item on the screen and verify that the information is correct.

If you need to make any changes, you can go back to any previous screen if you need to make any changes, either by using the **Back** button or by selecting the screen in the navigation pane.

Click **Update** to execute the domain extension.

In the Configuration Progress screen, click **Next** when it finishes.

For more information about the options on this screen, see Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 17 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen shows the following items about the domain that you just configured, including:

- Domain Location
- Administration Server URL

Make a note of both these items, because you need them later; you need the domain location to access the scripts used to start the Administration Server, and you need the Administration Server URL to access the WebLogic Remote Console and Oracle Enterprise Manager Fusion Middleware Control.

Click **Finish** to dismiss the Configuration Wizard.

Task 18 Start the Administration Server

Start the Administration Server to ensure the changes that you have made to the domain have been applied.

Update Certificates for New Frontend Addresses

This section contains information about certificates for new frontend addresses.

 **Note:**

About Certificates for the domain extension.

Since the BAM, SOA, and WSM servers use the same listen addresses (different port), there is no need to create new certificates and update stores. Also, since the BAM cluster uses the same frontend address as the SOA cluster, there is no need to update the OHS certificates and update trusted keystores. No action is required.

Update the WebLogic Servers Security Settings

This section contains information about WebLogic Servers security settings.

Follow the steps described in the [Updating the WebLogic Servers Security Settings](#) and update SSL settings for the WLS_BAM1 and WLS_BAM2 servers.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the BAM instances, and you have restarted the Administration Server on SOAHOST1, you must then propagate the domain changes to the domain directories and machines.

The following table summarizes the steps required to propagate the changes to all the domain directories and machines.

Task	Description	More Information
Pack up the Extended Domain on SOAHOST1	Use the <code>pack</code> command to create a new template jar file that contains the new BAM Servers configuration. When you pack up the domain, create a template jar file called <code>soadomaintemplateExtBAM.jar</code> .	Packing Up the Extended Domain on SOAHOST1
Unpack the Domain in the Managed Servers Directory on SOAHOST1*	Unpack the template jar file in the Managed Servers directory on SOAHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1
Unpack the Domain on SOAHOST2	Unpack the template jar file in the Managed Servers directory on the SOAHOST2 local storage.	Unpacking the Domain on SOAHOST2

*If you are configuring Oracle BAM on separate hosts, then you would unpack the domain on BAMHOST1 and BAMHOST2, rather than on SOAHOST1 and SOAHOST2.

Adding the Enterprise Deployment Administration User to the Oracle BAM Administration Group

Before you validate the Oracle BAM configuration on the Managed Server, add the enterprise deployment administration user (`weblogic_soa`) to the `BAMAdministrator` group.

To perform this task, refer to [Configuring Roles for Administration of an Enterprise Deployment](#).

Starting and Validating the WLS_BAM1 Managed Server

After extending the domain, restarting the Administration Server, and propagating the domain to the other hosts, start the newly configured BAM servers.

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

```
https://admin.example.com:445/em
```
2. Log in to Fusion Middleware Control by using the Administration Server credentials.
3. In the Target Navigation pane, expand the domain to view the Managed Servers in the domain.
4. Select only the WLS_BAM1 Managed Server, and click **Start Up** on the Oracle WebLogic Server toolbar.

Note:

BAM Servers depend on the policy access service to be functional, so the WSM-PM Managed Servers in the domain need to be up and running and reachable before the BAM servers are started.

5. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_BAM1 Managed Server is up and running.
6. To verify that the BAM software is configured properly:

- a. Enter the following URL in the browser:

```
https://SOAHOST1:7006/bam/composer
```

The login screen for BAM's composer appears.

If you configured Oracle BAM on separate host computers, enter `BAMHOST1` in the URL, rather than `SOAHOST1`.

- b. Enter the `weblogic_soa` login credentials.

The BAM Composer screen appears.

 **Note:**

To validate the server URLs, disable (set to blank) the front-end host until you have completed the configuration for the web tier. If you do not disable the front-end host, all requests fail because they are redirected to the front-end address.

7. Enter the following URL:

```
https://SOAHOST1:7006/inspection.wsil/
```

If you configured Oracle BAM on separate host computers, enter BAMHOST1 in the URL, rather than SOAHOST1.

You should see an XML response with a list of links.

8. Enter the following URL in the browser:

```
https://SOAHOST1:7006/bam/cqservice/
```

If you configured Oracle BAM on separate host computers, enter BAMHOST1 in the URL, rather than SOAHOST1.

You should get a message in the browser indicating *BAM CQService is running*.

Starting and Validating the WLS_BAM2 Managed Server

After you start the WLS_BAM2 managed server, you must verify that the server status is reported as *Running* in the Remote Console and access the URLs to verify the status of the servers.

1. Log in to Fusion Middleware Control by using the Administration Server credentials.
2. In the Target Navigation pane, expand the domain to view the Managed Servers in the domain.
3. Select only the WLS_BAM2 Managed Server, and click **Start Up** on the Oracle WebLogic Server tool bar.
4. When the startup operation is complete, navigate to the Domain home page and verify that the WLS_BAM2 Managed Server is up and running. Access the equivalent URLs for the WLS_BAM2:

```
https://SOAHOST2:7006/bam/composer
```

The login screen for BAM's composer appears. Enter the login credentials. The BAM composer's menu is displayed.

5. Enter the following URL:

```
https://SOAHOST2:7006/inspection.wsil/
```

You should see a response with a list of links.

6. Enter the following URL in the browser:

```
http://SOAHOST2:7006/bam/cqservice/
```

You should get a message in the browser indicating *BAM CQService is running*.

 **Note:**

If you configured Oracle BAM on separate host computers, enter *BAMHOST2* in the URL, rather than *SOAHOST2*.

Modifying the Upload and Stage Directories to an Absolute Path

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. See [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).

Configuring the Web Tier for the Extended Domain

Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.

- [Configuring Oracle HTTP Server for the WLS_BAM Managed Servers](#)
Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle BAM requests correctly to the Oracle BAM software on the Oracle SOA Suite cluster.

Configuring Oracle HTTP Server for the WLS_BAM Managed Servers

Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle BAM requests correctly to the Oracle BAM software on the Oracle SOA Suite cluster.

Note that these instructions assume that you are configuring Oracle BAM on the same host as Oracle SOA Suite. If you use separate hosts for Oracle BAM, you must modify the `WebLogicCluster` parameter in the Oracle HTTP Server configuration files to reference the `BAMHOST` computers, rather than the `SOAHOST` computers.

To enable Oracle HTTP Server to route requests to Oracle BAM:

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

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

2. Add the following directives inside the `<VirtualHost>` tag in the `soa_vh.conf` file:

```
<Location /bam/composer >
  WLSRequest ON
  WebLogicCluster SOAHOST1:7006,SOAHOST2:7006
</Location>

<Location /OracleBAMWS>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7006,SOAHOST2:7006
</Location>

<Location /oracle/bam/>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7006,SOAHOST2:7006
</Location>
```

3. Log in to `WEBHOST2` and change directory to the following location so that you can update the configuration file for the second Oracle HTTP Server instance (`ohs2`):

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf
```

4. Open the `soa_vh.conf` file and add the BAM directives to the `<VirtualHost>` tag.
5. Restart the Oracle HTTP Server instances on `WEBHOST1` and `WEBHOST2`.

Validating Access to Oracle BAM Through the Hardware Load Balancer

Verify that Oracle BAM URLs are successfully routing requests from the hardware load balancer to the Oracle HTTP Server instances to the Oracle BAM software in the middle tier.

You can also use this procedure to test the failover of the Managed Servers where Oracle BAM is configured.

To verify the URLs:

1. While the `WLS_BAM1` Managed Server is running, stop the `WLS_BAM2` Managed Server by using the Oracle WebLogic Remote console.
2. Access the following URL and verify the HTTP response as indicated in [Starting WLS_BAM1 Managed Server](#):

```
https://soa.example.com/bam/composer
```

3. Access the following URL to be sure the software is running as expected:

```
https://soa.example.com/oracle/bam/server
```

4. Start `WLS_BAM2` from the Oracle WebLogic Remote console.
5. Stop `WLS_BAM1` from the Oracle WebLogic Remote console.
6. Access the URL again, and verify that the HTTP response is still valid, as indicated in [Starting and Validating the WLS_BAM2 Managed Server](#).

Replacing Connect Strings with the Appropriate TNS Alias

Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

For more information about how to use TNS alias in your Datasources, see [Using TNS Alias in Connect Strings](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extend a domain or at another logical point. Create a backup after you verify that the installation so far

is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see [Performing Backups and Recoveries in the SOA Enterprise Deployments](#).

Extending the Domain with Oracle B2B

The procedures explained in this chapter guide you through the process of extending the enterprise deployment domain to include Oracle B2B.

The Oracle B2B and Healthcare distribution includes the software required to configure Oracle B2B or Oracle SOA for Healthcare.

 **Note:**

Healthcare in a WebLogic Domain option was deprecated in Fusion Middleware 12.2.1.4.0 and is removed from 14.1.2 release. Therefore, the chapter titled *Extending the Domain with Oracle SOA Suite for Healthcare Integration* is no longer included in this guide. This chapter covers the procedure to include B2B only.

- [Variables Used When Configuring Oracle B2B](#)
As you perform the tasks in this chapter, you reference the directory variables that are listed in this section.
- [Support for Reference Configuration in Oracle B2B](#)
Oracle B2B supports both SOA classic domains and SOA Reference Configuration domains. There is a specific B2B template when B2B is added to a classic SOA domain (the B2B classic template), and a specific B2B template when B2B is added to a Reference Configuration SOA domain (B2B Reference Configuration template).
- [Prerequisites for Extending the SOA Domain to Include Oracle B2B](#)
Before you extend the current domain, ensure that your existing deployment meets the prerequisites specified in this section.
- [Installing Oracle B2B for an Enterprise Deployment](#)
Use the following sections to install the Oracle Fusion Middleware Infrastructure software in preparation for configuring a new domain for an enterprise deployment.
- [Running the Configuration Wizard to Extend for Oracle B2B](#)
To extend the domain to include Oracle B2B, refer to the following sections.
- [Propagating the Extended Domain to the Domain Directories and Machines](#)
After you have extended the domain with the B2B instances, and have restarted the Administration Server on SOAHOST1, you must propagate the domain changes to the domain directories and machines.
- [Starting the B2B Suite Components](#)
For configuration changes and start scripts to be effective, you must start the WLS_SOA server to which B2B has been added. Since B2B extends an already existing SOA system, the Administration Server and the respective Node Managers are already running in SOAHOST1 and SOAHOST2.
- [Updating the B2B Instance Identifier for Transports](#)
To set up File, FTP, or Email transports in a high availability environment, set the `b2b.HAInstance` property to true.

- [Configuring the Web Tier for the Extended Domain](#)
Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.
- [Adding the B2BAdmin Role to the SOA Administrators Group](#)
Before you validate the Oracle B2B configuration on the Managed Servers, add the B2BAdmin administration role to the enterprise deployment administration group (SOA Administrators).
- [Validating Access to Oracle B2B Through the Load Balancer](#)
Use the following steps to verify that the appropriate routing and failover is working from the load balancer to the HTTP Server instances to the B2B Suite Components on the Oracle SOA Suite Managed Server.
- [Replacing Connect Strings with the Appropriate TNS Alias](#)
Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.
- [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 Oracle B2B

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

The values for several directory variables are defined in [File System and Directory Variables Used in This Guide](#).

- *ORACLE_HOME*
- *ASERVER_HOME*
- *MSERVER_HOME*
- *WEB_DOMAIN_HOME*
- *JAVA_HOME*

In addition, you reference the following virtual IP (VIP) addresses that are defined in [Physical and Virtual IP Addresses Required by the Enterprise Topology](#):

- ADMINVHN

Actions in this chapter are performed on the following host computers:

- SOAHOST1
- SOAHOST2
- WEBHOST1
- WEBHOST2

Support for Reference Configuration in Oracle B2B

Oracle B2B supports both SOA classic domains and SOA Reference Configuration domains. There is a specific B2B template when B2B is added to a classic SOA domain (the B2B classic

template), and a specific B2B template when B2B is added to a Reference Configuration SOA domain (B2B Reference Configuration template).

Prerequisites for Extending the SOA Domain to Include Oracle B2B

Before you extend the current domain, ensure that your existing deployment meets the prerequisites specified in this section.

- Back up the installation. If you have not yet backed up the existing Fusion Middleware Home and domain, Oracle recommends backing it up now.
To back up the existing Fusion Middleware Home and domain, see [Performing Backups and Recoveries in the SOA Enterprise Deployments](#).
- There is an existing `WL_HOME` and `SOA_ORACLE_HOME` (binaries) installed in previous chapters on a shared storage and available from `SOAHOST1` and `SOAHOST2`.
- Node Manager, Admin Server, SOA Servers, and WSM Servers exist and have been configured as described in previous chapters to run a SOA system.
- You do not need to run RCU to load additional schemas for B2B, these are part of the SOA repository and were loaded into the DB in the SOA chapter.
- You do not need to create an additional cluster because B2B components are added to the previously created `SOA_cluster`.

Installing Oracle B2B for an Enterprise Deployment

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

- [Starting the Oracle B2B and Healthcare Installer on SOAHOST1](#)
- [Navigating the Oracle B2B Installation Screens](#)
- [Installing the Software on Other Host Computers](#)
- [Verifying the B2B or Healthcare Installation](#)

Starting the Oracle B2B and Healthcare Installer on SOAHOST1

To start the installation program, perform the following steps.

1. Log in to `SOAHOST1`.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

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

In this example:

- Replace `JAVA_HOME` with the environment variable or actual JDK location on your system.
- Replace `distribution_file_name` with the actual name of the distribution jar file.

Note that if you download the distribution from the Oracle Technology Network (OTN), then the jar file is typically packaged inside a downloadable compressed file.

To install the software required for the B2B domain, the distribution that you want to install is **fmw_14.1.2.0.0_b2bhealthcare.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 Oracle B2B Installation Screens

[Table 17-1](#) provides description of each installation program screen.

Table 17-1 Oracle B2B Install Screens


Screen	Description
Installation Inventory Setup	<p>On UNIX operating systems, if this is the first time you are installing any Oracle product on this host, this screen appears. 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.</p> <p>For more information about the central inventory, see Understanding the Oracle Central Inventory in <i>Installing Software with the Oracle Universal Installer</i>.</p>
	<div style="border: 1px solid #0070C0; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>Oracle recommends that you configure the central inventory directory on the products shared volume. Example: <code>/u01/oracle/products/oraInventory</code></p> <p>You may also need to execute the <code>createCentralInventory.sh</code> script as root from the <code>oraInventory</code> folder after the installer completes.</p> </div>
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.

Table 17-1 (Cont.) Oracle B2B Install Screens

Screen	Description
Installation Location	Use this screen to specify the location of your Oracle home directory. For more information about Oracle Fusion Middleware directory structure, see <i>Selecting Directories for Installation and Configuration</i> in <i>Planning an Installation of Oracle Fusion Middleware</i> .
Installation Type	Use this screen to select the type of installation and consequently, the products and feature sets that you want to install. Select B2B NOTE: The topology in this document does not include the examples, Oracle strongly recommends that you do not install the examples into a production environment.
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the following documents in <i>Roadmap for Verifying Your System Environment</i> in <i>Installing and Configuring the Oracle Fusion Middleware Infrastructure</i> .
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. For more information about silent or command-line installation, see <i>Using the Oracle Universal Installer in Silent Mode</i> in <i>Installing Software with the Oracle Universal Installer</i> . Click Install to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click Next when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click Finish to dismiss the installer.

Installing the Software on Other Host Computers

If you have configured a separate shared storage volume or partition for SOAHOST2, then you must also install the software on SOAHOST2. For more information, see [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

Note that the location where you install the Oracle home (which contains the software binaries) varies, depending upon the host. To identify the proper location for your Oracle home directories, refer to the guidelines in [File System and Directory Variables Used in This Guide](#).

Verifying the B2B or Healthcare Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- [Reviewing the Installation Log Files](#)

- [Viewing the Contents of Your Oracle Home](#)

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see *Understanding Installation Log Files* in *Installing Software with the Oracle Universal Installer*.

Viewing the Contents of Your Oracle Home

You can also view the contents of your Oracle home by using the `viewInventory` script. See *Viewing the contents of an Oracle home* in *Installing Software with the Oracle Universal Installer*.

Running the Configuration Wizard to Extend for Oracle B2B

To extend the domain to include Oracle B2B, refer to the following sections.

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens for Oracle B2B](#)

Starting the Configuration Wizard

Note:

SSL store customizations were added to the `setUserOverridesLate.sh` in the domain creation chapter. Any customizations added to this file are preserved when a domain is extended and are carried over to remote servers when using the **pack** and **unpack** commands.

However, if you added any additional customizations to the `setDomainEnv.sh` script in the domain (such as custom libraries, JAVA command line options for starting the servers or environment variables), those will be overwritten by the configuration wizard when you extend the domain. Add all the startup parameters that apply to all servers in a domain to the `setUserOverridesLate.sh` file. This will preserve them across extensions.

To start the Configuration Wizard:

1. From the WebLogic Remote Console, stop any managed servers that are modified by this domain extension. Managed Servers that are not effected can remain on-line.

Note:

This specific domain extension for Oracle B2B component modifies the `WLS_SOAn` managed servers. Be sure to shut down these Managed Servers.

2. Verify the status of the managed servers, and then stop the Administration Server.
3. Navigate to the following directory and start the WebLogic Server Configuration Wizard.

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

Navigating the Configuration Wizard Screens for Oracle B2B

Follow the instructions in this section to extend the domain for Oracle B2B.

Note:

This procedure assumes that you are extending an existing domain. If your needs do not match the instructions given in the procedure, ensure that you make your selections accordingly, or refer to the supporting documentation for additional details.

Domain creation and configuration includes the following tasks:

- [Task 1, Selecting the Domain Type and Domain Home Location](#)
- [Task 2, Selecting the Configuration Template](#)
- [Task 3, Providing the GridLink Oracle RAC Database Connection Details](#)
- [Task 4, Testing the JDBC Connections](#)
- [Task 5, Selecting Advanced Configuration](#)
- [Task 6, Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Task 7, Writing Down Your Domain Home and Administration Server URL](#)
- [Task 8, Start the Administration Server](#)

Task 1 Selecting the Domain Type and Domain Home Location

On the Configuration Type screen, select **Update an existing domain**.

In the Domain Location field, select the value of the `ASERVER_HOME` variable, which represents the complete path to the Administration Server domain home you created in [Creating the Initial Infrastructure Domain for an Enterprise Deployment](#).

For more information about the directory location variables, see [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 Template

On the Templates screen, make sure **Update Domain Using Product Templates** is selected, then select the following templates:

- **Oracle B2B Reference Configuration [soa]**

In addition, the following additional templates should already be selected, because they were used to create the initial domain and extend it to SOA:

- Basic Weblogic Server Domain [wlserver]
- Oracle SOA Suite Reference Configuration [soa]

- Oracle Enterprise Manager [em]
- Oracle WSM Policy Manager [oracle_common]
- Oracle JRF [oracle_common]
- WebLogic Coherence Cluster Extension [wlserver]

 **Note:**

If you are extending B2B on a Classic SOA domain, you need to select the B2B classic extension template. To select the B2B Classic extension template:

- Oracle B2B - 14.1.2.0 [soa]

 **Tip:**

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

Task 3 Providing the GridLink Oracle RAC Database Connection Details

All fields are pre-populated because you already configured the domain to reference the Fusion Middleware schemas that are required for the Infrastructure domain. B2B uses the existing data sources for SOA and no new data sources need to be added to the domain.

 **Note:**

Any custom data sources that were created before the extension will show up before this screen. Check the Datasources row and click **Next**. The test data source screen will verify its validity.

Click **Next**.

Task 4 Testing the JDBC Connections

On the Test JDBC Data Sources screen, confirm that all connections were successful. The connections are tested automatically. The Status column displays the results. If all connections are not successful, click Previous to return to the previous screen and correct your entries.

Click **Next** when all the connections are successful.

Task 5 Selecting Advanced Configuration

To complete domain configuration for the topology, do not select any additional options on the Advanced Configuration screen and Click **Next**. B2B applications and required artifacts will be targeted automatically to the existing SOA servers

Task 6 Reviewing Your Configuration Specifications and Configuring the Domain

The Configuration Summary screen contains the detailed configuration information for the domain you are about to extend. Review the details of each item on the screen and verify that the information is correct.

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. Click **Update** to execute the domain extension. 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 7 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen will show the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you will need them later; the domain location is needed to access the scripts used to start the Administration Server, and the URL is needed to access the Administration Server.

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

Task 8 Start the Administration Server

Start the Administration Server to ensure the changes you have made to the domain have been applied.

Propagating the Extended Domain to the Domain Directories and Machines

After you have extended the domain with the B2B instances, and have restarted the Administration Server on SOAHOST1, you must propagate the domain changes to the domain directories and machines.

The following table summarizes the steps required to propagate the changes to all the domain directories and machines.

Task	Description	More Information
Pack up the Extended Domain on SOAHOST1	Use the <code>pack</code> command to create a new template jar file that contains the new BAM Servers configuration. When you pack up the domain, create a template jar file called <code>soadomaintemplateExtB2B.jar</code> .	Packing Up the Extended Domain on SOAHOST1
Unpack the Domain in the Managed Servers Directory on SOAHOST1	Unpack the template jar file in the Managed Servers directory on SOAHOST1 local storage.	Unpacking the Domain in the Managed Servers Domain Directory on SOAHOST1
Unpack the Domain on SOAHOST2	Unpack the template jar file in the Managed Servers directory on the SOAHOST2 local storage.	Unpacking the Domain on SOAHOST2

Starting the B2B Suite Components

For configuration changes and start scripts to be effective, you must start the WLS_SOA server to which B2B has been added. Since B2B extends an already existing SOA system, the Administration Server and the respective Node Managers are already running in SOAHOST1 and SOAHOST2.

To start the added B2B components, start the SOA managed servers:

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

```
https://ADMINVHN:9002/em
```

 **Note:**

If you have already configured web tier, use `https://admin.example.com:445/em`.

In this example:

Replace `ADMINVHN` with the host name assigned to the ADMINVHN Virtual IP address in [Identifying and Obtaining Software Downloads for an Enterprise Deployment](#).

Port `9002` is the typical Administration port used for the Remote 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.

2. In the Domain Structure window, expand the **Environment** node, then select **Servers**. The Summary of Servers page appears.
3. Click the **Control** tab.
4. Select **WLS_SOA1** from the Servers column of the table.

 **Note:**

SOA servers depend on the policy access service to be functional. This dependency implies that the WSM-PM servers in the domain need to be reachable before the SOA servers are started.

5. Click **Start**.
6. Repeat steps 2 through 5 for WLS_SOA2.

Updating the B2B Instance Identifier for Transports

To set up File, FTP, or Email transports in a high availability environment, set the `b2b.HAInstance` property to true.

To do this follow these steps:

1. Log in to Oracle Enterprise Manager Fusion Middleware Control with the user name and password specified for the domain administration.

2. Display the Target Navigation pane, by clicking the target navigation icon near the left top corner of the screen.
3. In the navigation tree, expand **SOA**, and then right click the `soa-infra(server_name)`, and select the **SOA Administration**, and then **B2B Server Properties** from the context menu.
If there are multiple `soa-infra (server_name)`, add the property only once.
4. Click **More B2B Configuration Properties**.
B2BConfig b2b should already be selected.
5. Click the **Operations** tab.
6. Click **addProperty** in the list on the right.
7. In the Key field enter **b2b.HAInstance**.
8. In the value field enter **true**.
This property is stored in MDS and needs to be created only once for the cluster.
9. Click **Invoke**.

After you define high availability properties, you can view them on the Attributes tab. To view the properties, click the **Attributes** tab and then click **Properties**. Expand the Element nodes in the Value table to verify the property names and values.

Configuring the Web Tier for the Extended Domain

Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.

- [Configuring Oracle HTTP Server for Oracle B2B](#)
Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle B2B requests correctly to the Oracle B2B software on the Oracle SOA Suite cluster.

Configuring Oracle HTTP Server for Oracle B2B

Make the following modifications to the Oracle HTTP Server instance configuration files to ensure that the Oracle HTTP Server instances in the web tier can route Oracle B2B requests correctly to the Oracle B2B software on the Oracle SOA Suite cluster.

To enable Oracle HTTP Server to route requests to Oracle B2B Console and to Oracle B2B services:

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

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs1/moduleconf
```

2. Add the following directives inside the `<VirtualHost>` tag in the `soa_vh.conf` file:

 **Note:**

The WebLogicCluster directive needs only a sufficient number of redundant server:port combinations to guarantee initial contact in case of a partial outage. The actual total list of cluster members is retrieved automatically upon first contact with any given node.

```
# B2B
<Location /b2bconsole>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# B2B
<Location /b2b/services>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>

# B2B
<Location /b2b/httpreceiver>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004
</Location>
```

3. Restart the `ohs1` instance on `WEBHOST1`.
4. Log in to `WEBHOST2` and copy the `soa_vh.conf` file to the configuration directory for the second Oracle HTTP Server instance (`ohs_2`):

```
cd $WEB_DOMAIN_HOME/config/fmwconfig/components/OHS/ohs2/moduleconf
```

5. Edit the `soa_vh.conf` file to change any references to `WEBHOST1` to `WEBHOST2`.
6. Restart the `ohs2` instance on `WEBHOST2`.

Adding the B2BAdmin Role to the SOA Administrators Group

Before you validate the Oracle B2B configuration on the Managed Servers, add the B2BAdmin administration role to the enterprise deployment administration group (SOA Administrators).

To perform this task, refer to [Configuring Roles for Administration of Oracle SOA Suite Products](#).

Validating Access to Oracle B2B Through the Load Balancer

Use the following steps to verify that the appropriate routing and failover is working from the load balancer to the HTTP Server instances to the B2B Suite Components on the Oracle SOA Suite Managed Server.

1. Enter the following URL to access the Oracle B2B Console through the load balancer:

```
https://soa.example.com/b2bconsole
```

2. Log in by using `weblogic_soa` user. You should see the Oracle B2B Partner, Agreement, and Profile screen.
3. Enter the following URL to access the Oracle B2B Web services endpoint:

`https://soa.example.com/b2b/services`

You see the links to the different B2B endpoints test.

Replacing Connect Strings with the Appropriate TNS Alias

Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

For more information about how to use TNS alias in your Datasources, see [Using TNS Alias in Connect Strings](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

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](#).

Configuring Oracle Managed File Transfer in an Enterprise Deployment

The procedures explained in this chapter guide you through the process of adding Oracle Managed File Transfer to your enterprise deployment.

- [About Oracle Managed File Transfer](#)
Oracle Managed File Transfer (MFT) provides a standards-based file gateway. It features design, deployment, and monitoring of file transfers by using a web-based design-time console that includes transfer prioritization, file encryption, scheduling, and embedded FTP and SFTP servers.
- [Variables Used When Configuring Managed File Transfer](#)
The procedures for installing and configuring Managed File Transfer reference use a series of variables that you can replace with the actual values used in your environment.
- [Synchronizing the System Clocks](#)
Verify that the system clocks on each host computer are synchronized.
- [Prerequisites for Creating the Managed File Transfer Domain](#)
Before you create the Managed File Transfer domain, ensure that your existing deployment meets the following prerequisites.
- [Installing the Software for an Enterprise Deployment](#)
The procedure to install the software for an enterprise deployment is explained in this section.
- [Creating the Managed File Transfer Database Schemas](#)
Before you can configure an Managed File Transfer domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.
- [Creating the Managed File Transfer Domain for an Enterprise Deployment](#)
- [Download and Configure Weblogic Remote Console](#)
This section describes how to download and configure the WebLogic Remote Console.
- [Configuring SSL Certificates for the Domain](#)
This section describes how to configure SSL certificates for the domain.
- [Configuring a Per Host Node Manager for an Enterprise Deployment](#)
For specific enterprise deployments, Oracle recommends that you configure a per-host Node Manager, as opposed to the default per-domain Node Manager.
- [Configuring the Domain Directories and Starting the Servers on MFTHOST1](#)
After the domain is created and the node manager is configured, you can then configure the additional domain directories and start the Administration Server and the Managed Servers on *MFTHOST1*.
- [Configuring Web Services Manager](#)
This section describes how to configure Web Services Manager.
- [Propagating the Domain and Starting the Servers on MFTHOST2](#)
After you start and validate the Administration Server and WLS_MFT1 Managed Server on *MFTHOST1*, you can then perform the following tasks on *MFTHOST2*.
- [Modifying the Upload and Stage Directories to an Absolute Path](#)

- [Configuring the Web Tier for the Extended Domain](#)
Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.
- [Validating the Managed File Transfer URLs Through the Load Balancer](#)
This section describes how to validate the configuration of Oracle HTTP Server and to verify that the hardware load balancer routes requests through the OHS instances to the application tier.
- [Configuring and Enabling the SSH-FTP Service for Managed File Transfer](#)
The Oracle Managed File Transfer enterprise deployment topology is based on the Secure File Transfer Protocol (SFTP) for file transfer. SFTP is a separate protocol, packaged with SSH and designed to work similar to FTP, but over a secure connection.
- [Creating a New LDAP Authenticator and Provisioning Users for Managed File Transfer](#)
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.
- [Replacing Connect Strings with the Appropriate TNS Alias](#)
Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.
- [Backing Up the Configuration](#)
It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

About Oracle Managed File Transfer

Oracle Managed File Transfer (MFT) provides a standards-based file gateway. It features design, deployment, and monitoring of file transfers by using a web-based design-time console that includes transfer prioritization, file encryption, scheduling, and embedded FTP and sFTP servers.

For more information about Oracle MFT, see *Understanding Oracle Managed File Transfer in Using Oracle Managed File Transfer*.

- [About Managed File Transfer in an Enterprise Deployment](#)
- [Characteristics of the Managed File Transfer Domain](#)

About Managed File Transfer in an Enterprise Deployment

Managed File Transfer runs in its own domain, separate from other components, such as Oracle SOA Suite, Oracle Service Bus, and Business Activity Monitoring. Typically, you create the domain and configure the Managed Servers for Managed File Transfer in a single configuration wizard session.

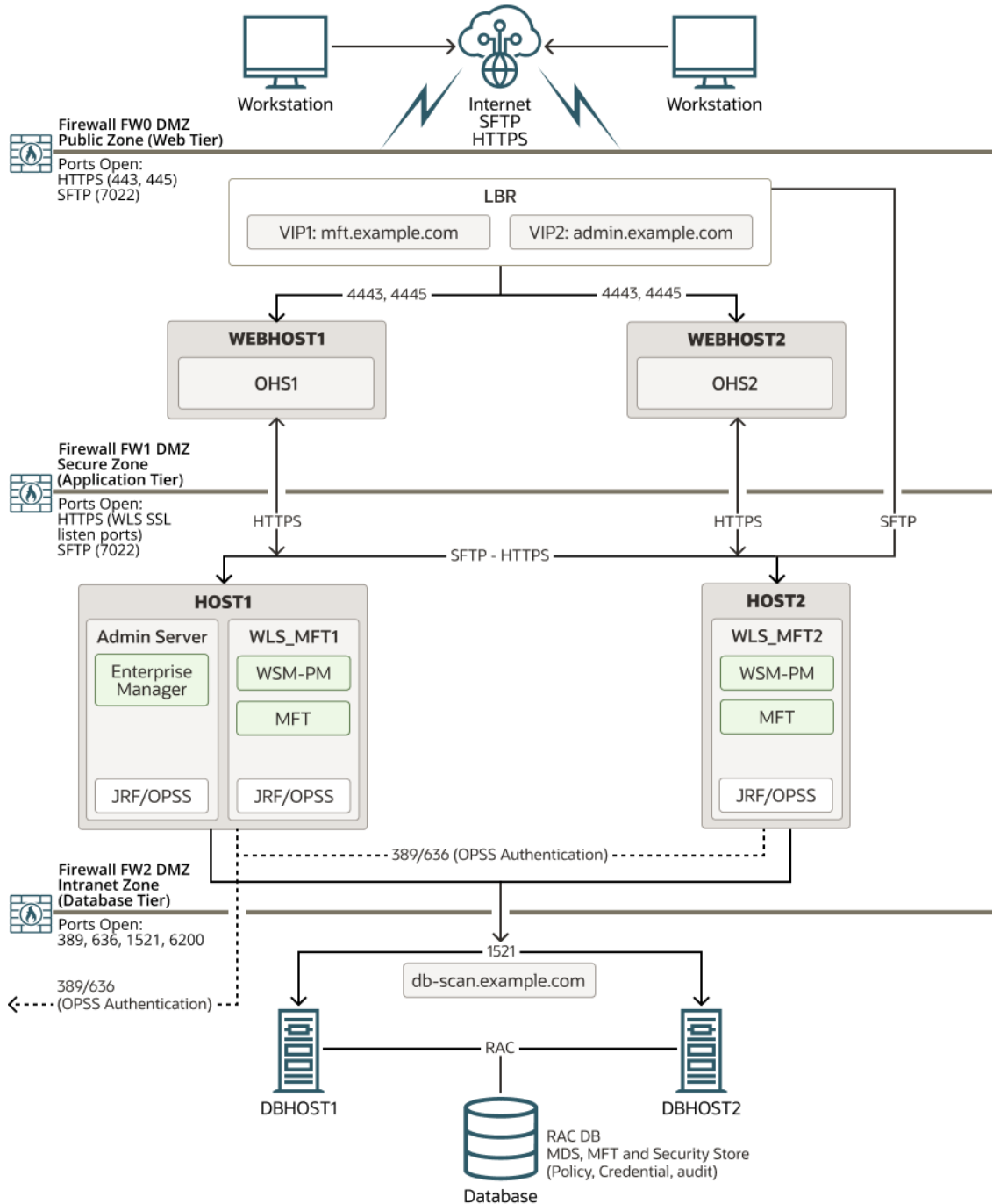
Managed File Transfer uses Oracle Web Services Manager (OWSM), and runs the OWSM services on the same servers as the Managed File Transfer applications.

[Figure 18-1](#) illustrates the Managed File Transfer deployment topology.

For a description of the standard elements shown in the diagram, see [Understanding the Typical Enterprise Deployment Topology Diagram](#).

For a description of the elements shown in the diagram, see [Understanding the Primary Oracle SOA Suite Topology Diagrams](#).

Figure 18-1 Managed File Transfer Topology



The Managed File Transfer domain can be configured on the same host as other FMW components. For this reason, Oracle recommends that you use a per host Node Manager configuration. In this configuration, a single Node Manager can control different domains on the same machine. See [Configuring a Per Host Node Manager for an Enterprise Deployment](#).

Characteristics of the Managed File Transfer Domain

The following table lists some of the key characteristics of the domain that you are about to create. 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](#).

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 Oracle Web Services Manager, which is deployed to the same servers as Managed File Transfer	Using Oracle Web Services Manager in the Application Tier
Uses the Load Balancer for routing SFTP requests from clients to MFT servers.	Configuring Oracle Load Balancer for SFTP Services
Uses a single Configuration Wizard session to configure the Infrastructure and Managed File Transfer software on the Managed File Transfer Managed Servers.	Creating the Managed File Transfer Domain for an Enterprise Deployment
Uses a per host Node Manager configuration.	About the Node Manager Configuration in a Typical Enterprise Deployment
Requires a separately installed LDAP-based authentication provider.	Understanding OPSS and Requests to the Authentication and Authorization Stores

Variables Used When Configuring Managed File Transfer

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

The following directory location variables are used in these procedures:

- `ORACLE_HOME`
- `JAVA_HOME`
- `NM_HOME`
- `ASERVER_HOME`
- `MSERVER_HOME`
- `WEB_ORACLE_HOME`
- `WEB_DOMAIN_HOME`

See [File System and Directory Variables Used in This Guide](#).

In addition, you reference the following virtual IP (VIP) address that are defined in [Reserving the Required IP Addresses for an Enterprise Deployment](#):

- `ADMINVHN`

Actions in this chapter are performed on the following host computers:

- `APPHOST1`

- APPHOST2
- WEBHOST1
- WEBHOST2

 **Note:**

Note that for this chapter, APPHOST1 and APPHOST2 provide a more generic variable for the application tier hosts. This is because, depending upon the domain you are creating, the host name variable varies.

Synchronizing the System Clocks

Verify that the system clocks on each host computer are synchronized.

Oracle recommends the use of the Network Time Protocol (NTP). See [Configuring a Host to Use an NTP \(time\) Server](#).

To verify the time synchronization, query the NTP service by running the `chronyc -n tracking` command on each host.

Sample output:

```
$chronyc -n tracking
Reference ID : A9FEA9FE (169.254.169.254)
Stratum : 3
Ref time (UTC) : Tue Jan 14 15:28:01 2025
System time : 0.000043127 seconds fast of NTP time
Last offset : +0.000034640 seconds
...
```

Prerequisites for Creating the Managed File Transfer Domain

Before you create the Managed File Transfer domain, ensure that your existing deployment meets the following prerequisites.

- You must have a supported database to install the schemas used by the MFT domain.
- Verify that you have installed a supported JDK.
- You must have an existing Oracle home where you have installed the Oracle Fusion Middleware Infrastructure software binaries. This must be a dedicated Oracle home for the Managed File Transfer domain. The Oracle home is typically on shared storage and is available from MFTHOST1 and MFTHOST2. See [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

Note that you should not configure the Infrastructure domain, only install the Oracle Fusion Middleware Infrastructure software.

To create the Infrastructure Oracle home, see [Installing the Oracle Fusion Middleware Infrastructure on SOAHOST1](#).

- Back up the installation. If you have not yet backed up the existing Fusion Middleware Home, Oracle recommends backing it up now.

To back up the existing Fusion Middleware Home and domain, see [Performing Backups and Recoveries in the SOA Enterprise Deployments](#).

- If you have not done so already, verify that the system clocks on each host computer are synchronized. You can do this by running the date command as simultaneously as possible on the hosts in each cluster.

Alternatively, there are third-party and open-source utilities that you can use for this purpose.

Installing the Software for an Enterprise Deployment

The procedure to install the software for an enterprise deployment is explained in this section.

- [Starting the Managed File Transfer Installer on MFTHOST1](#)
- [Navigating the Installation Screens When Installing Managed File Transfer](#)
- [Installing the Software on Other Host Computers](#)
- [Verifying the Installation](#)

Starting the Managed File Transfer Installer on MFTHOST1

To start the installation program:

1. Log in to MFTHOST1.
2. Go to the directory where you downloaded the installation program.
3. Launch the installation program by invoking the `java` executable from the JDK directory on your system, as shown in the example below.

```
JAVA_HOME/bin/java -jar Installer File Name
```

Be sure to replace the JDK location in these examples with the actual JDK location on your system.

Replace *Installer File Name* with the name of the actual installer file for your product listed in [Identifying and Obtaining Software Distributions for an Enterprise Deployment](#).

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

Navigating the Installation Screens When Installing Managed File Transfer

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

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

Screen	Description
Welcome	This screen introduces you to the product installer.
Auto Updates	Use this screen to automatically search My Oracle Support for available patches or automatically search a local directory for patches that you have already downloaded for your organization.

Screen	Description
Installation Location	Use this screen to specify the location of your Oracle home directory. This Oracle home should already contain Oracle Fusion Middleware Infrastructure. For more information about Oracle Fusion Middleware directory structure, see <i>Selecting Directories for Installation and Configuration</i> in <i>Planning an Installation of Oracle Fusion Middleware</i> .
Prerequisite Checks	This screen verifies that your system meets the minimum necessary requirements. If there are any warning or error messages, you can refer to one of the documents in the Roadmap for Verifying Your System Environment section in <i>Installing and Configuring the Oracle Fusion Middleware Infrastructure</i> .
Installation Summary	Use this screen to verify the installation options that you selected. Click Install to begin the installation.
Installation Progress	This screen allows you to see the progress of the installation. Click Next when the progress bar reaches 100% complete.
Installation Complete	Review the information on this screen, then click Finish to dismiss the installer.

Installing the Software on Other Host Computers

If you have configured a separate shared storage volume or partition for SOAHOST2, then you must also install the software on SOAHOST2. For more information, see [Shared Storage Recommendations When Installing and Configuring an Enterprise Deployment](#).

Note that the location where you install the Oracle home (which contains the software binaries) varies, depending upon the host. To identify the proper location for your Oracle home directories, refer to the guidelines in [File System and Directory Variables Used in This Guide](#).

Verifying the Installation

After you complete the installation, you can verify it by successfully completing the following tasks.

- [Reviewing the Installation Log Files](#)
- [Checking the Directory Structure for Managed File Transfer](#)

Reviewing the Installation Log Files

Review the contents of the installation log files to make sure that no problems were encountered. For a description of the log files and where to find them, see *Understanding Installation Log Files* in *Installing Software with the Oracle Universal Installer*.

Checking the Directory Structure for Managed File Transfer

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

Use the `ls --format=single-colum` command to check the list of directory and sub-directories in the `/u01/oracle/products/fmw` directory:

```
ls --format=single-colum $ORACLE_HOME
bin
```



```
coherence
em
install
inventory
jlib
lib
mft
OPatch
opmn
oracle_common
oraInst.loc
osb
oui
soa
wlserver
```

For more information about the directory structure you should see after installation, see [What are the Key Oracle Fusion Middleware Directories?](#) in *Understanding Oracle Fusion Middleware*.

Creating the Managed File Transfer Database Schemas

Before you can configure an Managed File Transfer domain, you must install the required schemas in a certified database for use with this release of Oracle Fusion Middleware.

- [Starting the Repository Creation Utility \(RCU\)](#)
- [Navigating the RCU Screens to Create the Managed File Transfer Schemas](#)
- [Verifying Schema Access](#)

Starting the Repository Creation Utility (RCU)

To start the Repository Creation Utility (RCU):

1. Navigate to the `$ORACLE_HOME/oracle_common/bin` directory on your system.

```
cd $ORACLE_HOME/oracle_common/bin
```

2. Make sure that the `JAVA_HOME` environment variable is set to the location of a certified JDK on your system. The location should be up to but not including the `bin` directory. For example, if your JDK is located in `/u01/oracle/products/jdk`:

On UNIX operating systems:

```
export JAVA_HOME=/u01/oracle/products/jdk
```

3. Start RCU:

On UNIX operating systems:

```
./rcu
```

 **Note:**

If your database has Transparent Data Encryption (TDE) enabled, and you want to encrypt your tablespaces that are created by the RCU, provide the `-encryptTablespace true` option when you start RCU.

This defaults 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 Managed File Transfer Schemas

Schema creation involves the following tasks:

- [Task 1, Introducing RCU](#)
- [Task 2, Selecting a Method of Schema Creation](#)
- [Task 3, Providing Database Connection Details](#)
- [Task 4, Specifying a Custom Prefix and Selecting Schemas](#)
- [Task 5, Specifying Schema Passwords](#)
- [Task 6, Verifying the Tablespaces for the Required Schemas](#)
- [Task 7, Creating Schemas](#)
- [Task 8, Reviewing Completion Summary and Completing RCU Execution](#)

Task 1 Introducing RCU

Click **Next**.

Task 2 Selecting a Method of Schema Creation

If you have the necessary permission and privileges to perform DBA activities on your database, select **System Load and Product Load**. This procedure assumes that you have the necessary privileges.

If you do not have the necessary permission or privileges to perform DBA activities in the database, you must select **Prepare Scripts for System Load** on this screen. This option generates a SQL script, which can be provided to your database administrator to create the required schema. See Understanding System Load and Product Load in *Creating Schemas with the Repository Creation Utility*.

Click **Next**.

Task 3 Providing Database Connection Details

Provide the database connection details for RCU to connect to your database.

1. As Database Type, select **Oracle Database enabled for edition-based redefinition**.

 **Note:**

Oracle Database enabled for edition-based redefinition (EBR) is recommended to support Zero Down Time upgrades. For more information, see <https://www.oracle.com/database/technologies/high-availability/ebr.html>.

2. In the **Host Name** field, enter the SCAN address of the Oracle RAC Database.
3. Enter the **Port** number of the RAC database scan listener, for example 1521.
4. Enter the RAC **Service Name** of the database.
5. Enter the **User Name** of a user that has permissions to create schemas and schema objects, for example `SYS`.
6. Enter the **Password** of the user name that you provided in step 4.
7. If you have selected the `SYS` user, ensure that you set the role to `SYSDBA`.
8. Click **Next** to proceed, then click **OK** on the dialog window confirming that the connection to the database was successful.

Task 4 Specifying a Custom Prefix and Selecting Schemas

On this page, do the following:

1. Choose **Create new prefix**, and then enter the prefix that you want to use for the Managed File Transfer schemas. A unique schema prefix is required because you are creating a new domain for Managed File Transfer.
2. From the list of schemas, select the **Managed File Transfer** schema.

The following dependent schemas are selected automatically:

- **Common Infrastructure Services**
- **Oracle Enterprise Scheduler**
- **Oracle Platform Security Services**
- **User Messaging Service**
- **Audit Services**
- **Audit Services Append**
- **Audit Services Viewer**
- **Metadata Services**
- **Weblogic Services**

The custom prefix is used to logically group these schemas together for use in this domain only; you must create a unique set of schemas for each domain as schema sharing across domains is not supported.

Tip:

For more information about custom prefixes, see Understanding Custom Prefixes in *Creating Schemas with the Repository Creation Utility*.
For more information about how to organize your schemas in a multi-domain environment, see Planning Your Schema Creation in *Creating Schemas with the Repository Creation Utility*.

Click **Next** to proceed, then click **OK** on the dialog window confirming that prerequisite checking for schema creation was successful.

Task 5 Specifying Schema Passwords

Specify how you want to set the schema passwords on your database, then specify and confirm your passwords. Ensure that the complexity of the passwords meet the database

security requirements before you continue. RCU proceeds at this point even if you do not meet the password policies. Hence, perform this check outside RCU itself.

 **Tip:**

You must make a note of the passwords you set on this screen; you need them later on during the domain creation process.

Click **Next**.

Task 6 Verifying the Tablespaces for the Required Schemas

On the Map Tablespaces screen, review the information, and then click **Next** to accept the default values.

Click **OK** in the confirmation dialog box.

Task 7 Creating Schemas

Review the summary of the schemas to be loaded, and click **Create** to complete schema creation.

 **Note:**

If failures occurred, review the listed log files to identify the root cause, resolve the defects, and then use RCU to drop and recreate the schemas before you continue.

Task 8 Reviewing Completion Summary and Completing RCU Execution

When you reach the Completion Summary screen, verify that all schema creations have been completed successfully, and then click **Close** to dismiss RCU.

Verifying Schema Access

Verify schema access by connecting to the database as the new schema users created by the RCU. Use SQL*Plus or another utility to connect, and provide the appropriate schema names and passwords entered in the RCU.

For example:

 **Note:**

If the database is a pluggable database (PDB), the appropriate tns alias that points to the PDB must be used in the sqlplus command.

```
sqlplus FMW1412_MFT/<mft_schema_password>
```

```
SQL*Plus: Release 23.0.0.0.0 - Production on Tue Jun 11 10:54:41 2024  
Version 23.4.0.24.05
```

```
Copyright (c) 1982, 2024, Oracle. All rights reserved.
```

```
Last Successful login time: Tue Jun 11 2024 10:52:21 +00:00
```

```
Connected to:  
Oracle Database 23ai EE Extreme Perf Release 23.0.0.0.0 - Production
```

Version 23.4.0.24.05

SQL>

Creating the Managed File Transfer Domain for an Enterprise Deployment

You create a separate Managed File Transfer domain by using the Fusion Middleware Configuration Wizard.

- [Starting the Configuration Wizard](#)
- [Navigating the Configuration Wizard Screens to Configure the MFT Domain](#)

Starting the Configuration Wizard

Start the Configuration Wizard as the first step to create the MFT enterprise deployment domain.

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

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

Navigating the Configuration Wizard Screens to Configure the MFT Domain

Follow the instructions in this section to create and configure the domain for MFT, with static clusters.

Domain creation and configuration includes the following tasks.

- [Task 1, Selecting the Domain Type and Domain Home Location](#)
- [Task 2, Selecting the Configuration Templates](#)
- [Task 3, Configuring High Availability Options](#)
- [Task 4, Selecting the Application Home Location](#)
- [Task 5, Configuring the Administrator Account](#)
- [Task 6, Specifying the Domain Mode and JDK](#)
- [Task 7, Specifying the Database Configuration Type](#)
- [Task 8, Specifying JDBC Component Schema Information](#)
- [Task 9, Providing the GridLink Oracle RAC Database Connection Details](#)
- [Task 10, Testing the JDBC Connections](#)
- [Task 11, Specifying the Keystore](#)
- [Task 12, Selecting Advanced Configuration](#)
- [Task 13, Configuring the Administration Server Listen Address](#)
- [Task 14, Configuring Node Manager \(Per Host\)](#)
- [Task 15, Configuring Managed Servers](#)

- [Task 16, Configuring a Cluster](#)
- [Task 17, Assigning Server Templates](#)
- [Task 18, Configuring Dynamic Servers](#)
- [Task 19, Assigning Managed Servers to the Cluster](#)
- [Task 20, Configuring Coherence Clusters](#)
- [Task 21, Creating Machines](#)
- [Task 22, Assigning Servers to Machines](#)
- [Task 23, Reviewing Your Configuration Specifications and Configuring the Domain](#)
- [Task 24, Writing Down Your Domain Home and Administration Server URL](#)

Task 1 Selecting the Domain Type and Domain Home Location

You must select a Domain home directory location, optimally outside the Oracle home directory.

Oracle recommends that you locate your Domain home in accordance with the directory structure in *What Are the Key Oracle Fusion Middleware Directories?* in *Understanding Oracle Fusion Middleware*, where the Domain home is located outside the Oracle home directory. This directory structure helps avoid issues when you need to upgrade or reinstall software.

To specify the Domain type and Domain home directory:

1. On the Configuration Type screen, select **Create a new domain**.
2. In the **Domain Location** field, specify your Domain home directory.

For more information about this screen, see 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 Managed File Transfer [mft]**

Selecting this template automatically selects the following dependencies:

- Oracle Enterprise Manager
- Oracle B2B Client
- Oracle JRF
- Oracle WSM Policy Manager
- WebLogic Coherence Cluster Extension

For more information about the options on this screen, see Templates in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 3 Configuring High Availability Options

This screen appears for the first time when you create a cluster that uses Automatic Service Migration or JDBC stores or both. After you select HA Options for a cluster, all subsequent clusters that are added to the domain by using the Configuration Wizard, automatically apply HA options (that is, the Configuration Wizard creates JDBC stores and configures ASM for them).

On the High Availability Options screen:

- Select **Enable Automatic Service Migration** with **Database Basis**.

- Set **JTA Transaction Log Persistence** to **JDBC TLog Store**.
- Set **JMS Server Persistence** to **JMS JDBC Store**.

 **Note:**

Oracle recommends that you use JDBC stores, which leverage the consistency, data protection, and high availability features of an Oracle database and makes resources available for all the servers in the cluster. So, the Configuration Wizard steps assume that the JDBC persistent stores are used along with Automatic Service Migration.

When you choose JDBC persistent stores, additional unused File Stores are automatically created but are not targeted to your clusters. Ignore these File Stores. If, for any reason, you want to use File Stores, you can retain the default values for TLOGs and JMS persistent store options in this screen and configure them in a shared location later. See [Task 12, Selecting Advanced Configuration](#). Shared location is required to resume JMS and JTA in a failover scenario.

You can also configure TLOGs and JMS persistent stores manually in a post step. For information about the differences between JDBC and File Stores, and for specific instructions to configure them manually, see [Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment](#).

Click **Next**.

Task 4 Selecting the Application Home Location

On the Application Location screen, specify the value of the `APPLICATION_HOME` variable, as defined in [File System and Directory Variables Used in This Guide](#).

For more information about the options on this screen, see 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 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.
- In the **Enable or Disable Default Ports for You Domain** field, use the default values provided for Production Mode:
 - Ensure that **Enable Listen Ports (non-SSL Ports)** is NOT checked.
 - Ensure that **Enable SSL Listen Ports** is checked.
 - Ensure that **Enable Administration Port (SSL Port)** is checked.

 **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*.

- Select **Oracle Hotspot** JDK in the JDK field.


 **Note:**

Ensure that it points to the folder where you have installed the JDK. See [Installing the JDK Software](#).

Task 7 Specifying the Database Configuration Type

On the Database Configuration Type screen:

- Select **RCU Data** to activate the fields on this screen.
 The **RCU Data** option instructs the Configuration Wizard to connect to the database and Service Table (STB) schema to automatically retrieve schema information for the schemas needed to configure the domain.
- Verify that the **Vendor** is Oracle and the **Driver** is *Oracle's Driver (Thin) for Service Connections; Versions: Any.
- Verify that **Connection Parameters** is selected.

 **Note:**

If you select **Manual Configuration** on this screen, you must manually fill in the parameters for your schema on the JDBC Component Schema screen.

After you select **RCU Data**, fill in the fields as shown in the following table:

Field	Description
DBMS/Service	Enter the service name for the Oracle RAC database where you install the product schemas. For example: soaedg.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.
Host Name	Enter the Single Client Access Name (SCAN) Address for the Oracle RAC database, which you entered in the <i>Enterprise Deployment Workbook</i> .
Port	Enter the port number on which the database listens. For example, 1521.

Field	Description
Schema Owner Schema Password	Enter the user name and password to connect to the database's Service Table schema. This is the schema user name and password that was specified for the Service Table component on the Schema Passwords screen in RCU. See Creating the Database Schemas . The default user name is <code>prefix_STB</code> , where <code>prefix</code> is the custom prefix that you have defined in RCU.

Click **Get RCU Configuration** when you finished specifying the database connection information. The following output in the Connection Result Log indicates that the operating succeeded:

```
Connecting to the database server...OK
Retrieving schema data from database server...OK
Binding local schema components with retrieved data...OK

Successfully Done.
```

Click **Next** if the connection to the database is successful. For more information about the **RCU Data** option, see Understanding the Service Table Schema in *Creating Schemas with the Repository Creation Utility*. For more information about the other options on this screen, see Datasource Defaults in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 8 Specifying JDBC Component Schema Information

Verify that the values on the JDBC Component Schema screen are correct for all schemas. The schema table should be populated, because you selected **Get RCU Data** on the previous screen. As a result, the Configuration Wizard locates the database connection values for all the schemas required for this domain.

At this point, the values are configured to connect to a single-instance database. However, for an enterprise deployment, you should use a highly available Real Application Clusters (RAC) database, as described in [Preparing the Database for an Enterprise Deployment](#).

In addition, Oracle recommends that you use an Active GridLink datasource for each of the component schemas. For more information about the advantages of using GridLink data sources to connect to a RAC database, see Database Considerations in the *High Availability Guide*.

To convert the data sources to GridLink:

1. Select all the schemas by selecting the checkbox 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 that is required to connect to the RAC database and component schemas, as shown in following table.

Element	Description and Recommended Value
Service Name	Verify that the service name for the Oracle RAC database is the appropriate. For example, <code>soaedg.example.com</code> .

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	These values are not required when you are using an Oracle 12c database or higher versions because the ONS list is automatically provided from the database to the driver.
Enable Fan	Verify that the Enable Fan check box is selected, so that the database can receive and process FAN events.

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

Task 10 Testing the JDBC Connections

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.

By default, the schema password for each schema component is the password you specified while creating your schemas. If you want different passwords for different schema components, manually edit them in the previous screen (JDBC Component Schema) by entering the password you want in the **Schema Password** column, against each row. After you specify the passwords, select the check box that correspond to the schemas that you changed the password in and test the connection again.

For more information about the other options on this screen, see Test Component Schema in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 11 Specifying the Keystore

Use the Keystore screen in the Configuration Wizard to specify details about the keystore to be used in the domain.

For a typical enterprise deployment, you can leave the default values. See Keystore in *Creating WebLogic Domains Using the Configuration Wizard*

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 add, delete, or modify the Settings for Server Templates, Managed Servers, Clusters, Virtual Targets, and Coherence.

 **Note:**

- If any of the above options are not available on the screen, then return to the Templates screen and ensure that you have selected the required templates for this topology.
- JDBC stores are recommended and selected in [Task 3, Configuring High Availability Options](#) so there is no need to configure File Stores.

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 about the purpose of using the ADMINVHN virtual host, see [Reserving the Required IP Addresses for an Enterprise Deployment](#).

3. In the **Configure Administration Server Ports** section, perform the following steps:
 - a. Leave the **Enable Listen Port** field unchecked. The **Listen Port** value will be disabled in grey.
 - b. Ensure the **Enable SSL Listen port** field is checked.
 - c. Leave the default value as 7002 in the **SSL Listen Port** field.
 - d. Leave the default value as 9002 in the **Administration Port**.
4. Leave the default value as Unspecified in the **Server Group**.

Task 14 Configuring Node Manager (Per Host)

Select **Manual Node Manager Setup** as the Node Manager type.

 **WARNING:**

You can ignore the warning in the bottom pane. This guide provides the required steps for the Manual Node Manager configuration.

 **Tip:**

For more information about the options on this screen, see Node Manager in *Creating WebLogic Domains Using the Configuration Wizard*.

For more information about per domain and per host Node Manager implementations, see [About the Node Manager Configuration in a Typical Enterprise Deployment](#).

For information about Node Manager configurations, see Configuring Node Manager on Multiple Machines in *Administering Node Manager for Oracle WebLogic Server*.

Task 15 Configuring Managed Servers

Use the Managed Servers screen to create the Managed Servers that are required in the Managed File Transfer domain.

1. Change the default server name to `WLS_MFT1` in the **Server name** column.
2. Click **Add** and repeat this process to create a second Managed Server named `WLS_MFT2`.
3. Use the information in [Table 18-1](#) to fill in the rest of the columns for each MFT Managed Server.

The Managed Server names suggested in this procedure (`WLS_MFT1` and `WLS_MFT2`) are referenced throughout this document; if you choose different names then be sure to replace them as needed.

For more information about the options on this screen, see Managed Servers in *Creating WebLogic Domains Using the Configuration Wizard*.

Server Name	Listen Address	Enable Listen Port	Listen Port	Enable SSL Port	SSL Listen Port	Administration Port	Server Groups
WLS_MFT1	MFTHOST1	Unchecked	Disabled	Checked	7010	9014	MFT-MGD-SVRS
WLS_MFT2	MFTHOST2	Unchecked	Disabled	Checked	7010	9014	MFT-MGD-SVRS

The selected server group ensures that the Managed File Transfer and Oracle Web Services Manager (OWSM) software is targeted to the Managed Server.

There is another server group called **MFT-MGD-SVRS-ONLY** that targets only MFT but not Oracle Web Services Manager (OWSM) to the server. This is typically used if you want to have Oracle Web Services Manager (OWSM) in a different server rather than with the MFT server.

The server groups target Fusion Middleware applications and services to one or more servers by mapping defined groups of application services to each defined server group. Any application services that are mapped to a given server group are automatically targeted to all servers that are assigned to that group. See Application Service Groups, Server Groups, and Application Service Mappings in *Domain Template Reference*.

Task 16 Configuring a Cluster

Use the Clusters screen to create a new cluster:

1. Click the **Add** button.
2. Specify `MFT_Cluster` in the **Cluster Name** field.
3. Leave the **Cluster Address** field blank.
4. Specify `mft.example.com` in the **Frontend Host** field.
5. Specify `0` as the **Frontend HTTP** port and `443` as the **Frontend HTTPS** port.

For more information about the options on this screen, see Clusters in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 17 Assigning Server Templates

Click **Next**.

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 **Calculated Machine Names** and **Calculated Listen Port** checkboxes on this screen are unchecked.
2. Confirm that the **Server Template** and **Dynamic Server Groups** selections are **Unspecified**.
3. Click **Next**.

Task 19 Assigning Managed Servers to the Cluster

Use the Assign Servers to Clusters screen to assign Managed Servers to the new cluster.

1. In the **Clusters** pane, select the cluster to which you want to assign the servers; in this case, `MFT_Cluster`.
2. In the **Servers** pane, assign `WLS_MFT1` to `MFT_Cluster` by doing one of the following:
 - Click once on `WLS_MFT1` to select it, then click on the right arrow to move it beneath the selected cluster (`MFT_Cluster`) in the Clusters pane.
 - or*
 - Double-click on `WLS_MFT1` to move it beneath the selected cluster (`MFT_Cluster`) in the clusters pane.
3. Repeat these steps to assign the `WLS_MFT2` Managed Server to `MFT_Cluster`.

For more information about the options on this screen, see 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`.

For Coherence licensing information, Oracle Coherence Products in *Oracle Fusion Middleware Licensing Information User Manual*.

Task 21 Creating Machines

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

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

Use the values in [Table 18-2](#) to define the Name and Node Manager Listen Address of each machine.
3. Verify the port in the **Node Manager Listen Port** field.

The port number `5556`, shown in this example, may be referenced by other examples in the documentation. Replace this port number with your own port number as needed.

 **Note:**

If you are installing on a host where additional domains were already configured, and you have already configured a per host Node Manager, then the address and port configured in this screen must be for the existing per host Node Manager.

Name	Node Manager Listen Address	Node Manager Type	Node Manager Listen Port
MFTHOST1	The value of the MFTHOST1 host name variable or MFTHOST1 alias. For example, <i>MFTHOST1.example.com</i> .	SSL	5556
MFTHOST2	The value of the MFTHOST2 host name variable or MFTHOST2 alias. For example, <i>MFTHOST2.example.com</i> .	SSL	5556
ADMINHOST	Enter the value of the ADMINVHN variable.	SSL	5556

For more information about the options on this screen, see *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 two Managed Servers 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_MFT1 Managed Server to the *MFTHOST1* machine.
- Assign the WLS_MFT2 Managed Server to the *MFTHOST2* machine.

For more information about the options on this screen, see *Assign Servers to Machines* in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 23 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.
For more information about the options on this screen, see Configuration Summary in *Creating WebLogic Domains Using the Configuration Wizard*.

Task 24 Writing Down Your Domain Home and Administration Server URL

The Configuration Success screen shows the following items about the domain you just configured:

- Domain Location
- Administration Server URL

You must make a note of both items as you need them later; the domain location is needed to access the scripts used to start the Administration Server.
Click **Finish** to dismiss the Configuration Wizard.

Download and Configure Weblogic Remote Console

This section describes how to download and configure the WebLogic Remote Console.

Note:

For the initial configuration steps required in this EDG, you will need access to the AdminServer listen address and administration port. Later on you will configure access from a frontend load balancer.

Perform the following steps to download and configure the WebLogic Remote Console:

1. Uninstall any previous versions of the WebLogic Remote Console from your computer.
2. Download the WebLogic Remote Console. Go to <https://github.com/oracle/weblogic-remote-console/releases> and download the installer from your operating system.
3. Run the installer.
4. Install the WebLogic Remote Console extension in the WebLogic Server domain. The WebLogic Remote Console extension provides additional functionality when using the WebLogic Remote Console to manage WebLogic domains.

Note:

This step is optional.

- a. Create a `management-services-ext` directory under the domain home.
 - b. Download the latest WebLogic Remote Console extension, `console-rest-ext-<version>.war`, from <https://github.com/oracle/weblogic-remote-console/releases> and save it inside the `management-services-ext` directory you created in the previous step. If you have an earlier version of the extension already downloaded, delete it and replace it with the latest version.
 - c. Reboot the Administration Server if it is already running.
5. Launch the WebLogic Remote Console application.

Example:

```
./weblogic-remote-console
```

In the next steps you must connect to the EDG domain provider using initially the Admin Servers listen address.

Configuring SSL Certificates for the Domain

This section describes how to configure SSL certificates for the domain.

- [Creating Certificates and Certificate Stores for the WebLogic Domain](#)
- [Adding Certificate Stores Location to the WebLogic Servers Start Scripts](#)
- [Update Server's Security Settings Using the Remote Console](#)
- [Configuring KSS with Per-domain CA](#)

Creating Certificates and Certificate Stores for the WebLogic Domain

The Enterprise Deployment Guide provides steps to configure a domain that uses SSL listen addresses for all Weblogic Managed Servers, Weblogic Administration Server and Node Managers in the application tier. To achieve this the required certificates for all servers, machines and NM listen addresses must be created and pointed to from the domain and Node Manager configuration.

In Oracle FMW 14.1.2.0, Oracle WebLogic allows the usage of a per-domain Certificate Authority (CA). In this model, the CertGen and ImportPrivateKey utilities are enhanced to use the domain's secret key to encrypt the passphrases and store them in the domain's DemoCerts.props file. A self-signed Demo CA is automatically created for the domain and it is used for signing certificates for the SSL listen addresses used in the EDG. Although in a real production system, standard CAs should be used, the per-domain CA model implements an SSL system using domain specific CA that provides a higher degree of protection than non-ssl configurations. If you want to use your own custom certificates, see [About Using Third Party SSL Certificates in the WebLogic and Oracle HTTP Servers](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Oracle recommends using a shared storage location (protected with the appropriate snapshot or file backup tooling) where all the different certificates and stores can be found by the different servers. Perform the following steps to generate an Identity store and a Trust Store that can be used for enabling SSL listeners in a Weblogic Server using a per-domain CA:

1. Download the `generate_perdomainCACERTS.sh` script in the maa github repo.

```
https://github.com/oracle-samples/maa/blob/main/1412EDG/generate\_perdomainCACERTS.sh
```

2. Run the script with the following arguments:
 - `WLS_DOMAIN_DIRECTORY`: Directory hosting the Weblogic Domain that the Administration Server uses.
 - `WL_HOME`: The directory within the Oracle home where the Oracle WebLogic Server software binaries are stored. Typically `/u01/oracle/products/fmw/wlserver`.
 - `KEYSTORE_HOME`: Directory where `appIdentity` and `appTrust` stores will be created.
 - `KEYPASS`: Password used for the weblogic administration user (will be reused for certs and stores).

Example:

```
./generate_perdomainCACERTS.sh $ASERVER_HOME $ORACLE_HOME/  
wlsserver $KEYSTORE_HOME <keypass>
```

The script will traverse the `WLS_DOMAIN_DIRECTORY/config/config.xml` to find all the listen addresses used in the domain, generate certificates for all of them, create a Truststore with the domain CA and the WLS CertGenCA, and import certificates into a new Identity store. The aliases used in the import will be the same as the hostname used as listen address. Both the trust store and the identity store will be placed in the `KEYSTORE_HOME` directory.

Run the following command to verify if the "domainca" entry is there as a `trustedCertEntry`:

```
keytool -list -keystore $KEYSTORE_HOME/appTrustKeyStore.pkcs12
```

Run the following command to verify if there is a `PrivateKeyEntry` for each listen address (the values for ADMINVHN, SOAHOST1 and SOAHOST2):

```
keytool -list -keystore $KEYSTORE_HOME/appIdentityKeyStore.pkcs12
```

Adding Certificate Stores Location to the WebLogic Servers Start Scripts

Once the Identity and Trust Stores are created for the domain some Java properties must be added to the WebLogic start scripts. These properties are added to the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Any customizations you add to this file are preserved during domain upgrade operations and are carried over to remote servers when using the pack and unpack commands.

- If you created the Identity and Trust Stores with the script `generate_perdomainCACERTS.sh`, as explained in [Creating Certificates and Certificate Stores for the WebLogic Domain](#), then the properties are automatically added to the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Just verify that the file exists and that the `EXTRA_JAVA_PROPERTIES` have been added.
- If you are using your own custom certificates, then manually create the file `setUserOverridesLate.sh` in `$ASERVER_HOME/bin`. Edit the file and add the variable `EXTRA_JAVA_PROPERTIES` to set the `javax.net.ssl.trustStore` and `javax.net.ssl.trustStorePassword` properties with the values used by your EDG system. For example:

```
EXTRA_JAVA_PROPERTIES="{EXTRA_JAVA_PROPERTIES}  
-Djavax.net.ssl.trustStore=/u01/oracle/config/keystores/  
appTrustKeyStore.pkcs12  
-Djavax.net.ssl.trustStorePassword=mypassword"  
export EXTRA_JAVA_PROPERTIES
```

Note:

The order of the extra java properties is relevant. In case that the same property is defined more than once, the later value is used. The custom values must be defined as in the example provided.

Update Server's Security Settings Using the Remote Console

- [Connecting to the Remote Console Using the Administration Server's Virtual Hostname as Provider](#)
- [Updating the WebLogic Servers Security Settings](#)

Connecting to the Remote Console Using the Administration Server's Virtual Hostname as Provider

The following procedure temporarily starts the Administration Server with the default start script so to enable you to perform these tasks. After you perform these tasks, you can stop this temporary session and use the Node Manager to start the Administration Server.

 **Note:**

For this Remote Console initial access to the Administration Server, it is required that the machine that runs the Remote Console can resolve and connect to the Admin Server's Listen Address. This can be done by starting the Remote Console directly in the node where the Admin Server runs or creating a tunnel to this address from the node where the remote Console is executed.

1. Using the following default start script to start the Administration Server:

- a. Change directory to the following directory:

```
cd $ASERVER_HOME/bin
```

- b. Run the start script:

```
./startWebLogic.sh
```

Monitor the terminal till the following message is displayed:

```
<Server state changed to RUNNING>
```

Also you must verify that the appropriate SSL listener is available, which can be confirmed with the a message like the following displayed in output:

```
<Server> <BEA-002613> <Channel "DefaultSecure" is now listening on  
XXXX:7002 for protocols iiops, t3s, ldaps, https.>
```

2. Create a new provider in the WebLogic Remote Console as follows:

- a. Download the domain's trust keystore to the host or laptop where you run the WebLogic Remote Console. For example, when using the per-domain CA steps in previous sections, this would be located at `$KEYSTORE_HOME/appTrustKeyStore.pkcs12`.
- b. Open the Remote Console and add the domain trust store to the remote console settings. Click **File > Settings** and enter the following values.
 - i. Trust Store type - jks

- ii. Trust Store Path - The path to the trust keystore file in the host where the Remote Console runs.
 - iii. Trust Store Key - Enter the password provided in the steps above for certificate creation.
 - iv. Check **Disable HostName verification** if you are using Demo certificates as described in the steps above.
- c. Using the Providers window in the Remote Console, create a new provider by selecting **Add Admin Server Connection Provider**.
- i. In the provider name, enter the name of `mftedg_domain_asvip`. This will identify the type of access.
 - ii. Enter the WebLogic Domain Administration username provided in the configuration wizard user name.
 - iii. Enter the password used for the domain creation.
 - iv. Use https protocol and the admin server listen address used in the configuration wizard as URL for access and specify port 9002.
For example, `https://ADMINVHN.example.com:9002`.
 - v. Check the **Make Insecure Connection** checkbox.



Note:

This provider should not be used once the front end and webtier are configured.

The Remote Console Home Window for the domain will be displayed.

Updating the WebLogic Servers Security Settings

Perform the following steps to update the WebLogic Servers Security Settings and Administration Port:

1. Access the Domain provider in the Remote Console and update the Administration Server and WebLogic Servers Security Settings:
 - a. Click **Edit Tree**.
 - b. Click **Environment > Servers > AdminServer**.
 - c. Click **Security** tab.
 - d. Change the keystores dropdown to **Custom Identity and Custom Trust**.
 - e. In **Custom Identity Keystore**, enter the fully qualified path to the identity keystore as follows:
`KEYSTORE_HOME/appIdentityKeyStore.pkcs12`
Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the [Table 7-2](#).
 - f. Set the **Custom Identity Keystore Type** to JKS.
 - g. In **Custom Identity Keystore Passphrase**, enter the password `Keystore_Password` you provided in the certificate generation steps.
 - h. In **Custom Trust Keystore**, enter the fully qualified path to the trust keystore.

```
KEYSTORE_HOME/appTrustKeyStore.pkcs12
```

Replace `KEYSTORE_HOME` with the value of the folder you use for storing keystore, as described in the [Table 7-2](#).

- i. Set the **Custom Trust Keystore Type** to JKS.
- j. In **Custom Trust Keystore Passphrase**, enter the password you provided as the **<keypass>** in the certificate generation steps.
- k. Click **Save**.
- l. Under **Security** settings, navigate to **SSL** tab.
- m. In the **Server Private Key Alias** field enter the alias provided in the certificate generation steps. If you used the certificate generation script this will be the same as the listen address used for the WLS server.
- n. In the **Server Private Key Pass Phrase** field, enter the password provided in the certificate generation steps. If you used the certificate generation script this will be the same as the keystore passphrase.
- o. Click **Save**.

The cart on the top right part of the screen will show **full** with a yellow bag inside.

- p. Click the Cart icon on the top right and select **Commit Changes**.

Repeat the above steps for each managed server in the domain changing the alias to match the alias used for the certificates.

2. Return to the terminal window where you started the Administration Server with the start script.
3. Press **Ctrl+C** to stop the Administration Server process.

Wait for the Administration Server process to end and for the terminal command prompt to appear.

4. Start the Administration Server again by using the following script:
 - a. Change directory to the following directory:

```
cd $ASERVER_HOME/bin
```

- b. Run the start script:

```
./startWebLogic.sh
```

- c. Monitor the output in the terminal till the following output is displayed.

```
<Server state changed to RUNNING>
```

Configuring KSS with Per-domain CA

For consistency purposes and to use a common CA all across the domain artifacts you may want to use the per-domain CA for KSS (store used by OPSS and other components in the WebLogic Infrastructure/JRF Domain).

Perform the following steps to import the domain CA certificate in the KSS trusted store:

1. Download the `import-domainca-into-kss.sh` script in the maa github repo <https://github.com/oracle-samples/maa/blob/main/1412EDG/import-domainca-into-kss.sh>.

2. Edit the script and customize the following variables according to your environment:

DOMAIN_HOME: Path to the WebLogic domain (ASERVER_HOME in this guide). For example, /u01/oracle/config/domains/mftedg_domain.

MW_HOME: The path to the FMW home. For example, /u01/oracle/products/fmw.

ADMINVHN: Administration Server's listen address. For example, *adminvhn.example.com*.

ADMINPORT: Administration Server's listen port. For example, 9002.

DOMAINUSER: Name of the administrator user for the WLS domain. For example, soaedgadmin.

TRUSTSTOREFILE: Location of the truststore used to connect though SSL to the Admin Server. For example, /u01/oracle/config/keystores/appTrustKeyStore.pkcs12.

3. Run the script with the following arguments:

- DOMAINPASS: WLS domain administrator user's password
- KEYPASS: Password for the truststore.

Example

```
./import-domainca-into-kss.sh adminpassword123 truststorepassword123
```

The script imports the per Domain CA certificate into KSS and assigns it to jps.

You can verify that the update was successful by inspecting the jps configuration files.

```
grep domainca $ASERVER_HOME/config/fmwconfig/jps-config.xml
```

The result of the command must be similar to the following example:

```
<property name="ca.key.alias" value="domainca-new-24-05-07-16-44-52"/>
```

4. Restart the Admin Server.

If Admin Server was started with the script, perform the following steps:

- a. Press **Ctrl+C** to stop the Administration Server process.
- b. Go to directory \$ASERVER_HOME/bin and run the following command:

```
./startWebLogic.sh
```

Configuring a Per Host Node Manager for an Enterprise Deployment

For specific enterprise deployments, Oracle recommends that you configure a per-host Node Manager, as opposed to the default per-domain Node Manager.

For more information about the advantages of a per host Node Manager, see [About the Node Manager Configuration in a Typical Enterprise Deployment](#).

- [Creating a Per Host Node Manager Configuration](#)
- [Starting the Node Manager on MFTHOST1](#)
- [Configuring the Node Manager Credentials](#)
- [Enrolling the Domain with NM](#)
- [Adding Truststore Configuration to Node Manager](#)

Creating a Per Host Node Manager Configuration

The step in configuring a per-host Node Manager is to create a configuration directory and two new node manager configuration files. You must also edit the default `startNodeManager.sh` file.

To create a per-host Node Manager configuration, perform the following tasks, first on MFTHOST1, and then on MFTHOST2:

1. Log in to MFTHOST1 and create a directory for the Node Manager configuration files :

For example:

```
mkdir -p /u02/oracle/config/nodemanager
```

Note that this directory should be on a local disk, because it is specific to the host. This directory location is known as the Node Manager home, and it is identified by the `NM_HOME` directory variable in examples in this guide.

2. Change directory to the Node Manager home directory:

```
cd $NM_HOME
```

3. Create a new text file called `nodemanager.properties` and add the values shown [Example: Contents of the nodemanager.properties File](#) in to this new file.

You must repeat similar configuration for the Node Managers in the other node of the domain (MFTHOST2, use the pertaining certificate alias).

Use the pertaining identity alias for the node that you are configuring. For example, `soahost1.example.com` in MFTHOST1 and `soahost2.example.com` in MFTHOST2.

For more information about the properties that you can add to the `nodemanager.properties` file, see [Node Manager Properties in Administering Node Manager for Oracle WebLogic Server](#).

In the `nodemanager.properties` file, you enable crash recovery for servers as a part of this configuration. See [Node Manager and System Crash Recovery in Administering Node Manager for Oracle WebLogic Server](#).

Example: Contents of the nodemanager.properties File

```
DomainsFile=/u02/oracle/config/nodemanager/nodemanager.domains
LogLevel=INFO
LogLimit=0
PropertiesVersion=14.1.2.0.0
AuthenticationEnabled=true
NodeManagerHome=/u02/oracle/config/nodemanager
#Include the specific JDK home
JavaHome=/u01/oracle/products/jdk
LogLevel=INFO
DomainsFileEnabled=true
StartScriptName=startWebLogic.sh
#Leave blank for listening on ANY
ListenAddress=
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
SecureListener=true
LogCount=1
StopScriptEnabled=false
```

```
QuitEnabled=false
LogAppend=true
StateCheckInterval=500
CrashRecoveryEnabled=true
StartScriptEnabled=true
LogFile=/u02/oracle/config/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityAlias=soahost1.example.com
CustomIdentityKeyStoreFileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12
CustomIdentityKeyStorePassPhrase=password
CustomIdentityPrivateKeyPassPhrase=password
```

Notice the values for `CustomIdentityAlias`. If you used the `generate_perdomainCACERTS.sh` script, this is the hostname used as listen address in the configuration wizard for the Node Manager Machine. If you created the certificates one by one, this would be the alias that you assigned to the certificate import for `MFTHOST1`. You must also provide the location of the IdentityStore generated in the previous steps and the password for the same.

4. Locate the `startNodeManager.sh` file in the following directory:

```
$WL_HOME/server/bin
```

5. Copy the `startNodeManager.sh` file to the Node Manager home directory.

```
cp $WL_HOME/server/bin/startNodeManager.sh $NM_HOME
```

6. Edit the new `startNodeManager.sh` file and add the `NODEMGR_HOME` property as follows:

```
NODEMGR_HOME="$NM_HOME"
```

In this example, replace `NM_HOME` with the actual path to the Node Manager home.

7. Locate the `stopNodeManager.sh` script in the `WL_HOME/server/bin` directory. Copy it to the Node Manager home directory. Edit the copied file and edit the `NODEMGR_HOME` property pointing to the node manager home (as it has been done for the `startNodemanager.sh` file):

```
NODEMGR_HOME="$NM_HOME"
```

In this example, replace `NM_HOME` with the actual path to the Node Manager home.

8. Create another new file in the Node Manager home directory, called `nodemanager.domains`.

The `nodemanager.domains` file provides additional security by restricting Node Manager client access to the domains listed in this file.

9. Perform steps 1 through 8 on `MFTHOST2`.
10. Add the following entries to the new `nodemanager.domains` files:

On `MFTHOST1`, add values for both the Administration Server domain home and the Managed Servers domain home:

```
mftedg_domain=MSERVER_HOME;ASERVER_HOME
```

 **Note:**

The path that is mentioned first (*MSERVER_HOME*) is considered as the `primaryDomainPath` and Managed Servers are run from this location.

On MFTHOST2, add the value for the Managed Servers domain home only:

```
mftedg_domain=MSERVER_HOME
```

In these examples, replace *ASERVER_HOME* and *MSERVER_HOME* with the values of the respective variables, as described in [File System and Directory Variables Used in This Guide](#).

Starting the Node Manager on MFTHOST1

After you manually set up the Node Manager to use a per-host Node Manager configuration, you can start the Node Manager on MFTHOST1, by using the `startNodeManager.sh` script.

To start the Node Manager on MFTHOST1:

1. Change directory to the Node Manager home directory:

```
cd $NM_HOME
```

2. Run the following command to start the Node Manager and send the output of the command to an output file, rather than to the current terminal shell:

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

3. Monitor the `nodemanager.out` file; make sure the NodeManager starts successfully. The output should eventually contain the following strings:

```
<INFO> <Upgrade> <Encrypting NodeManager property:
CustomIdentityKeyStorePassPhrase>
<INFO> <Upgrade> <Encrypting NodeManager property:
CustomIdentityPrivateKeyPassPhrase>
<Upgrade> <Saving upgraded NodeManager properties to '/u02/oracle/config/
nodemanager/nodemanager.properties'>
<INFO> <Loading domains file: /u02/oracle/config/nodemanager/
nodemanager.domains>
<INFO> <Loading identity key store: FileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12, Type=pkcs12, PassPhraseUsed=true>
<INFO> <Loaded NodeManager configuration properties from '/u02/oracle/
config/nodemanager/nodemanager.properties'>
<INFO> <14.1.2.0.0>
<INFO> <Server Implementation Class:
weblogic.nodemanager.server.NMServer$ClassicServer.>
<INFO> <Secure socket listener started on port 5556>
```

You must check that the plain text used for passwords in `nodemanager.properties` has now been encrypted:

```
[oracle@soalonhost1 keystores]$ cat /u02/oracle/config/nodemanager/
```



```

nodemanager.properties
#Tue Feb 06 11:53:10 GMT 2024
#Mon Feb 05 17:24:30 GMT 2024
DomainsFile=/u02/oracle/config/nodemanager/nodemanager.domains
LogLevel=0
PropertiesVersion=14.1.2.0.0
AuthenticationEnabled=true
NodeManagerHome=/u02/oracle/config/nodemanager
#Include the specific JDK home
JavaHome=/u01/oracle/products/jdk
LogLevel=INFO
DomainsFileEnabled=true
StartScriptName=startWebLogic.sh
#Leave blank for listening on ANY
ListenAddress=
NativeVersionEnabled=true
ListenPort=5556
LogToStderr=true
SecureListener=true
LogCount=1
StopScriptEnabled=false
QuitEnabled=false
LogAppend=true
StateCheckInterval=500
CrashRecoveryEnabled=true
StartScriptEnabled=true
LogFile=/u02/oracle/config/nodemanager/nodemanager.log
LogFormatter=weblogic.nodemanager.server.LogFormatter
ListenBacklog=50
KeyStores=CustomIdentityAndCustomTrust
CustomIdentityAlias=soahost1.example.com
CustomIdentityKeyStoreFileName=/u01/oracle/config/keystores/
appIdentityKeyStore.pkcs12
CustomIdentityKeyStorePassPhrase={AES256}EMvProCRfN7fyv3d8JcEnttTLyng9Su+U
VK5DGEmqmqDwLkpLz9nQFZ+fL1Bidc
CustomIdentityPrivateKeyPassPhrase={AES256}O5cEJD8WVYP3aRLp9KAbFZ3CGLyxmmIW
FX1YzVfJvPp11dc5RbMksAcsBLquKcWW

```

Configuring the Node Manager Credentials

Perform the following steps to set the Node Manager credentials using the Remote Console:

1. Access the Domain provider in the Remote Console.
2. Click **Edit Tree**.
3. Click **Environment > Domain > Security**.
4. Check the **Show Advanced Fields** field.
5. Set **Node Manager Username** to the same as the Weblogic Administrator, since this username will be used in other tasks mentioned in this guide.
6. Change the NM password. Ensure the **Node Manager password** is set to the same as the Weblogic Administrator since this password will be used in other tasks mentioned in this guide.
7. Click **Save**. The cart on the top right part of the screen will show **full** with a yellow bag inside.

8. Click the Cart Icon on the top right and select **Commit Changes**.

Enrolling the Domain with NM

Perform the following steps in a new terminal window to enroll the domain with Node manager.

Note:

You will be unable to connect to the Node Manager and use it to start the servers in the domain without performing this step.

1. Change directory to the following directory:

```
cd $ORACLE_COMMON_HOME/common/bin
```

2. Start the WebLogic Server Scripting Tool (WLST). In order to use the certificates created for the appropriate SSL handshake, the location of the stores and password of the same need to be provided to WLST. Use the following command or add these in a script that can be easily invoked:

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=CustomTrust -  
Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/  
keystores/appTrustKeyStore.pkcs12 -  
Dweblogic.security.CustomTrustKeyStorePassPhrase=storepassword"  
./wlst.sh
```

Note:

You must avoid including the password in the script.

3. Connect to the Administration Server by using the following WLST command:

```
connect('admin_user','admin_password','admin_url')
```

For example:

```
connect('weblogic','<password>','t3s://ADMINVHN:9002')
```

4. Use the `nmEnroll` command to enable the Node Manager to manage servers in a specified WebLogic domain.

```
nmEnroll('ASERVER_HOME')
```

For example:

```
nmEnroll('/u01/oracle/config/domains/mftedg_domain')
```

5. Generate startup properties for the Admin Server using the following WLST command:

```
nmGenBootStartupProps('AdminServer')
```

The `startup.properties` file is created in the following directory:

```
$ASERVER_HOME/servers/AdminServer/data/nodemanager/
```

 **Note:**

This step is optional and must be performed only if you want to customize any startup properties for the Administration Server.

Adding Truststore Configuration to Node Manager

It is required to add the corresponding truststore configuration for Node Manager communication with the different WebLogic Server listeners. To do this, edit Node Manager's start script `startNodeManager.sh` located at `$NM_HOME` and add the variable `JAVA_OPTIONS` to set the `javax.net.ssl.trustStore` and `javax.net.ssl.trustStorePassword` properties with the values used by your EDG system. For example:

```
export JAVA_OPTIONS="${JAVA_OPTIONS} -Djavax.net.ssl.trustStore=/u01/oracle/
config/keystores/appTrustKeyStore.pkcs12 -
Djavax.net.ssl.trustStorePassword=mypassword"
```

 **Note:**

If you have used the `generate_perdomainCACERTS.sh` script to generate certificates and stores, the `trustStorePassword` is the password provided as "KEYPASS" parameter to the script.

Configuring the Domain Directories and Starting the Servers on MFTHOST1

After the domain is created and the node manager is configured, you can then configure the additional domain directories and start the Administration Server and the Managed Servers on *MFTHOST1*.

- [Disabling the Derby Database](#)
- [Starting the Administration Server Using the Node Manager](#)
After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.

- [Validating the Administration Server](#)
Before you proceed with the configuration steps, validate that the Administration Server has started successfully by making sure that you have access to the Oracle WebLogic Remote 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 MFTHOST1](#)
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 MFTHOST1 and MFTHOST2. The domain directory on the local (or private) storage is used to run the Managed Servers.
- [Starting and Validating the WLS_MFT1 Managed Server on MFTHOST1](#)
After you have configured Node Manager and created the Managed Server domain directory, you can use Oracle Enterprise Manager Fusion Middleware Control to start the WLS_MFT1 Managed Server on MFTHOST1.

Disabling the Derby Database

Before you create the Managed Server directory and start the Managed Servers, disable the embedded Derby database, which is a file-based database, packaged with Oracle WebLogic Server. The Derby database is used primarily for development environments. As a result, you must disable it when you configure a production-ready enterprise deployment environment; otherwise, the Derby database process start 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. Complete steps 1 through 2 on each `ORACLE_HOME` for `MFTHOST1` and `MFTHOST2` if they use separate shared file systems.

Starting the Administration Server Using the Node Manager

After you have configured the domain and configured the Node Manager, you can start the Administration Server by using the Node Manager. In an enterprise deployment, the Node Manager is used to start and stop the Administration Server and all the Managed Servers in the domain.

To start the Administration Server by using the Node Manager:

1. Ensure that the Administration Server is stopped.
2. Start the WebLogic Scripting Tool (WLST):

```
export WLST_PROPERTIES=""
-Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/
keystores/appTrustKeyStore.pkcs12
-Dweblogic.security.CustomTrustKeyStorePassPhrase=password"
```

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

 **Note:**

The `weblogic.security.SSL.ignoreHostnameVerification=true` is required when using Demo certificates as the ones provided by the `generateCertificates` scripts. In an environment with formal CA and certificates, this flag should not be used.

3. Connect to Node Manager by using the Node Manager credentials:

```
nmConnect('nodemanager_username','nodemanager_password','ADMINVHN','5556',
domain_name','ASERVER_HOME','SSL')
```

Replace `ADMINVHN` and `ASERVER_HOME` with the values of the respective variables.

 **Note:**

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

```
ASERVER_HOME/config/nodemanager
```

4. Start the Administration Server:

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

 **Note:**

When you start the Administration Server, it attempts to connect to Oracle Web Services Manager for WebServices policies. It is expected that the WSM-PM Managed Servers are not yet started, and so, the following message appears in the Administration Server log:

```
<Warning><oracle.wsm.resources.policymanager>
<WSM-02141><Unable to connect to the policy access service due to
Oracle WSM policy manager host server being down.>
```

5. 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 Remote 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:

```
https://ADMINVHN:9002/em
```

You should be able to connect to the Admin Server from the Remote Console as before.

Creating a Separate Domain Directory for Managed Servers on MFTHOST1

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 MFTHOST1 and MFTHOST2. 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 cause by servers writing logs to shared storage. It is also faster to load classes and jars need from the domain directory, so any tmp directory 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. Log in to MFTHOST1 and run the `pack` command to create a template as follows:

```
cd $ORACLE_COMMON_HOME/common/bin

./pack.sh -managed=true
          -domain=ASERVER_HOME
          -template=complete_path/mftdomaintemplate.jar
          -template_name=create_domain_template
```

In this example:

- Replace *ASERVER_HOME* with the actual path to the domain directory that 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 need to reference this location when you copy or unpack the domain template jar file.
 - `mftdomaintemplate` is a sample name for the jar file that you are creating, which contains the domain configuration files.
 - `mft_domain_template` is the name assigned to the domain template file.
2. Make a note of the location of the template jar file that you created with the `pack` command.

 **Tip:**

For more information about the `pack` and `unpack` commands, see [Overview of the Pack and Unpack Commands in *Creating Templates and Domains Using the Pack and Unpack Commands*](#).

3. If you haven't already, create the recommended directory structure for the Managed Server domain on the MFTHOST1 local storage device.

Use the examples in [File System and Directory Variables Used in This Guide](#) as a guide.

4. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

```
cd $ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME \
            -overwrite_domain=true \
            -template=complete_path/mftdomaintemplate.jar \
            -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME
```

 **Note:**

The `-overwrite_domain` option in the `unpack` command allows unpacking a managed server template into an existing domain and existing applications directories. For any file that is overwritten, a backup copy of the original is created. If any modifications had been applied to the start scripts and ear files in the managed server domain directory, they must be restored after this `unpack` operation.

Additionally, to customize server startup parameters that apply to all servers in a domain, you can create a file called `setUserOverridesLate.sh` and configure it to, for example, add custom libraries to the WebLogic Server classpath, specify additional java command-line options for running the servers, or specify additional environment variables. Any customizations you add to this file are preserved during domain upgrade operations, and are carried over to remote servers when 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 `complete_path` with the complete path to the location where you created or copied the template jar file.
- `mftdomaintemplate.jar` is the name of the template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device.
- Replace `APPLICATION_HOME` with the complete path to the applications 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 Managed Server directory and verify that the domain configuration files were copied to the correct location on the MFTHOST1 local storage device.

Starting and Validating the WLS_MFT1 Managed Server on MFTHOST1

After you have configured Node Manager and created the Managed Server domain directory, you can use Oracle Enterprise Manager Fusion Middleware Control to start the WLS_MFT1 Managed Server on MFTHOST1.

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

```
https://ADMINVHN:9002/em
```

In this example:

- Replace *ADMINVHN* with the host name assigned to the ADMINVHN Virtual IP address in [Identifying and Obtaining Software Downloads for an Enterprise Deployment](#).
- Port 9002 is the typical Administration Port used for the Remote 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, see *Getting Started Using Oracle Enterprise Manager Fusion Middleware Control in [Administering Oracle Fusion Middleware](#)*.

2. Log in to Fusion Middleware Control by using the Administration Server credentials.
3. Select the **Servers** pane to view the Managed Servers in the domain.
4. Select only the **WLS_MFT1** Managed Server, and then click **Control** > **Start** on the tool bar.
5. To verify that the Managed Server is working correctly, open your browser and enter the following URLs:

```
https://MFTHOST1:7010/wsm-pm/
https://MFTHOST1:7010/mftconsole/
```


 **Note:**

- To validate the server URLs, disable (set to blank) the front-end host until you have completed the configuration for Oracle HTTP Server. If you do not disable the front-end host, all requests fail because they are redirected to the front-end address.

Enter the domain admin user name and password when prompted.

Configuring Web Services Manager

This section describes how to configure Web Services Manager.

- [Updating WebServices Domain Configuration](#)
- [Bootstrapping WSM](#)

Updating WebServices Domain Configuration

1. Log into the Fusion Middleware Control by using the administrator's account.
2. In the **Weblogic Domain** drop down menu, select **WebServices > WSM Domain Configuration**.
3. Click **Policies Access** tab.
4. Select the **Auto Discover** and **Use SSL Only** check boxes.
5. In the **SSL Setup** section, select **Oneway**.
6. In KeyStore Type, select JKS (Java Key Store).

 **Note:**

You must select JKS if you are using the certificates and stores created in previous steps.

7. In the Truststore Path enter the location of the truststore used in previous sections as follows:
`/u01/oracle/config/keystores/appTrustKeyStore.pkcs12`
8. In the **Key** field, enter a name to uniquely identify the password used for the truststore.
9. In the **password** field, enter the password used for the truststore in previous sections (same as domain admin).
10. Click **Apply**.

Bootstrapping WSM

In a new terminal window, perform the following steps to bootstrap WSMPM.

 **Note:**

If this task is not performed, the WSMPM does not work properly .

1. Change directory to the following directory as follows:

```
cd $ORACLE_COMMON_HOME/common/bin
```

2. Start the WebLogic Server Scripting Tool (WLST). To use the certificates created for the appropriate SSL handshake, the location of the stores and password of the same need to be provided to WLST. Use the following command or add these in a script that can be easily invoked (avoid including the password in the script):

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/
keystores/appTrustKeyStore.pkcs12
-Dweblogic.security.CustomTrustKeyStorePassPhrase=storepassword"
./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','<password>','t3s://ADMINVHN:9002')
```

4. Use the `setWSMBootstrapConfig` to enable WSM `pm.url`.

```
setWSMBootstrapConfig('domain_name','domainpath','ConfigManager','pm.url','
auto-ssl')
```

For example:

```
setWSMBootstrapConfig('mftedg_domain','/u01/oracle/config/domains/
mftedg_domain','ConfigManager','pm.url','auto-ssl')
```

Check that the appropriate entry is created in the `$ASERVER_HOME/config/fmwconfig/wsm-config.xml` for the domain. For example:

```
cat /u01/oracle/config/domains/mftedg_domain/config/fmwconfig/wsm-
config.xml
<?xml version="1.0" encoding="UTF-8"?>
<orares:properties xmlns:orares="http://xmlns.oracle.com/wsm/resources"
xmlns:wsp15="http://www.w3.org/ns/ws-policy" xmlns:orawsp="http://
schemas.oracle.com/ws/2006/01/policy" orares:type="CONFIGURATION"
orares:resource="/">
  <orares:property orares:category="ConfigManager" orares:name="pm.url">
    <orares:value>auto-ssl</orares:value>
```

```
</orares:property>  
</orares:properties>
```

Propagating the Domain and Starting the Servers on MFTHOST2

After you start and validate the Administration Server and WLS_MFT1 Managed Server on *MFTHOST1*, you can then perform the following tasks on *MFTHOST2*.

- [Unpacking the Domain Configuration on MFTHOST2](#)
- [Starting the Node Manager on MFTHOST2](#)
- [Starting and Validating the WLS_MFT2 Managed Server on MFTHOST2](#)

Unpacking the Domain Configuration on MFTHOST2

Now that you have the Administration Server and the first WLS_WSM1 Managed Server running on *MFTHOST1*, you can configure the domain on *MFTHOST2*.

1. Log in to *MFTHOST2*.
2. If you haven't already, create the recommended directory structure for the Managed Server domain on the MFTHOST2 storage device.

Use the examples in [File System and Directory Variables Used in This Guide](#) as a guide.

3. Make sure that the `mftedgdomaintemplate.jar` file is accessible to *MFTHOST2*.

For example, if you are using a separate shared storage volume or partition for *MFTHOST2*, then copy the template to the volume or partition mounted to *MFTHOST2*.

4. Run the `unpack` command to unpack the template in the domain directory onto the local storage, as follows:

```
cd $ORACLE_COMMON_HOME/common/bin  
  
./unpack.sh -domain=MSERVER_HOME \  
            -overwrite_domain=true \  
            -template=/complete_path/mftdomaintemplate.jar \  
            -log_priority=DEBUG \  
            -log=/tmp/unpack.log \  
            -app_dir=APPLICATION_HOME
```

In this example:

- Replace `MSERVER_HOME` with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain is unpacked.
- Replace `full_path` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device.
- Replace `APPLICATION_HOME` with the complete path to the Application directory for the domain on shared storage. For more information about the variables, 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 `MSERVER_HOME` directory and verify that the domain configuration files were copied to the correct location on the `MFTHOST2` local storage device.

Starting the Node Manager on MFTHOST2

After you manually set up the Node Manager to use a per host Node Manager configuration, you can start the Node Manager by using the following commands on `MFTHOST2`:

1. Change directory to the Node Manager home directory:

```
cd $NM_HOME
```

2. Run the following command to start the Node Manager and send the output of the command to an output file, rather than to the current terminal shell:

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

Starting and Validating the WLS_MFT2 Managed Server on MFTHOST2

Use the procedure that is explained in [Starting and Validating the WLS_MFT1 Managed Server on MFTHOST1](#) to start and validate the WLS_MFT2 Managed Server on `MFTHOST2`.

Modifying the Upload and Stage Directories to an Absolute Path

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. See [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).

Configuring the Web Tier for the Extended Domain

Configure the web server instances on the web tier so that the instances route to the proper clusters in the SOA domain.

- [Setting Frontend Addresses and WebLogic Plugin for the MFT Cluster and the Administration Server](#)
- [Configuring Oracle HTTP Server for Managed File Transfer](#)

Setting Frontend Addresses and WebLogic Plugin for the MFT Cluster and the Administration Server

As a security best practice oracle recommends setting a frontend address for the Administration Server and the MFT cluster. In the initial domain creation steps, since OHS and the frontend Load Balancer may have not been configured yet, the frontend setting is avoided to allow verifications using the individual server addresses. However, at this point and before

configuring OHS (and the frontend load balancer, if not done yet) it is required to add the pertaining addresses.

1. To set the front end for the Administration Server, use the WebLogic Remote Console as follows:
 - a. Click **Edit Tree**.
 - b. Click **Environment>Servers>AdminServer**.
 - c. Select the **Protocols** Tab and then select the **HTTP** tab.
 - d. As **Frontend Host**, enter the front end LBR address that is used to access Enterprise management and the Remote Console (*admin.example.com* in the example used in this guide).
 - e. Leave the **Frontend HTTP** port set to 0.
 - f. Enter the LBR's admin listener port (445) as **Frontend HTTPS port**.
 - g. Click **Save**.
 - h. Click the car icon at the top right to commit the changes.
2. The frontend of the MFT Cluster is configured during the domain creation. To verify the front end for the MFT Cluster, use the Remote Console as follows:
 - a. Click **Edit Tree**.
 - b. Click **Environment>Clusters>MFT_Cluster**.
 - c. Select the **HTTP** tab.
 - d. Verify that the value of the frontend name (for example, *mft.example.com*) is set as the **Frontend Host**.
 - e. Verify that the **Frontend HTTP** port set to 0.
 - f. Verify that the **Frontend HTTPS port** set to the appropriate value used by the LBR listener to access to MFT (443).
 - g. In case of any change, click **Save**.
 - h. Click the car icon at the top right to commit the changes.
 - i. Click **Save**.
 - j. Click the car icon at the top right to commit the changes.
3. Enable the proxy plugin for the domain using the WebLogic Remote Console as follows:
 - a. Click **Edit Tree**.
 - b. Click **Environment>Domains**.
 - c. Select **Web Application** tab.
 - d. Click the **Welogic Plugin Enable** button.
 - e. Click **Save**.
 - f. Click the car icon at the top right to commit the changes.

These changes requires a restart of the AdminServer and the `MFT_Cluster` to be effective (a notification appears in the WebLogic Remote Console about restart being required).

Configuring Oracle HTTP Server for Managed File Transfer

Follow the steps described in [Configuring Oracle HTTP Server for an Enterprise Deployment](#) for installing and configuring the OHS servers.

The only difference for MFT is that, instead of creating the `soainternal_vh.conf` file, you must create `mft_vh.conf` file with the following content, to route the requests to the MFT Cluster.

Ensure that you edit the file with the appropriate values for **Listen**, **ServerName**, **VirtualHost**, **SSLWallet** and **Location** directives:

File `mft_vh.conf`:

```
#####
# Oracle HTTP Server mod_ossll configuration file: ssl.conf      #
#####

# The Listen directive below has a comment preceding it that is used
# by tooling which updates the configuration.  Do not delete the comment.
#[Listen] OHS_SSL_PORT
Listen ohshost1.example.com:4443

##
## SSL Virtual Host Context
##
#[VirtualHost] OHS_SSL_VH
<VirtualHost ohshost1.example.com:4443>
ServerName mft.example.com:443
<IfModule ossll_module>
    # SSL Engine Switch:
    # Enable/Disable SSL for this virtual host.
    SSLEngine on

    # Client Authentication (Type):
    # Client certificate verification type and depth.  Types are
    # none, optional and require.
    SSLVerifyClient None

    # SSL Protocol Support:
    # Configure usable SSL/TLS protocol versions.
    SSLProtocol TLSv1.2 TLSv1.3

    # Option to prefer the server's cipher preference order
    SSLHonorCipherOrder on

    # SSL Cipher Suite:
    # List the ciphers that the client is permitted to negotiate.
    SSLCipherSuite
TLS_AES_128_GCM_SHA256,TLS_AES_256_GCM_SHA384,TLS_CHACHA20_POLY1305_SHA256,TLS
_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,T
LS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

    #Path to the wallet
    #SSLWallet "${ORACLE_INSTANCE}/config/fmwconfig/components/$
    {COMPONENT_TYPE}/instances/${COMPONENT_NAME}/keystores/default"
    SSLWallet "/u02/oracle/config/keystores/orapki/orapki-vh-WEBHOST1"
```

```

<FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
</FilesMatch>

<Directory "${ORACLE_INSTANCE}/config/fmwconfig/components/${
COMPONENT_TYPE}/instances/${COMPONENT_NAME}/cgi-bin">
    SSLOptions +StdEnvVars
</Directory>

BrowserMatch "MSIE [2-5]" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0

# Add the following directive to add HSTS
<IfModule mod_headers.c>
    Header always set Strict-Transport-Security "max-age=63072000; preload;
includeSubDomains"
</IfModule>

<Location /wsm-pm>
    WLSRequest ON
    WebLogicCluster MFTHOST1:7010,MFTHOST2:7010
</Location>
<Location /mftconsole>
    WLSRequest ON
    WebLogicCluster MFTHOST1:7010,MFTHOST2:7010
</Location>

</IfModule>

</VirtualHost>

```

Validating the Managed File Transfer URLs Through the Load Balancer

This section describes how to validate the configuration of Oracle HTTP Server and to verify that the hardware load balancer routes requests through the OHS instances to the application tier.

1. Verify that the server status is reported as **Running** in the WebLogic Remote Console. If the server is shown as **Starting** or **Resuming**, wait for the server status to change to **Started**. If another status (such as Admin or Failed) is reported, check the server output log files for errors.
2. Verify that you can access the following URL:

```

https://admin.example.com:445/em
https://mft.example.com:443/mftconsole

```

Configuring and Enabling the SSH-FTP Service for Managed File Transfer

The Oracle Managed File Transfer enterprise deployment topology is based on the Secure File Transfer Protocol (SFTP) for file transfer. SFTP is a separate protocol, packaged with SSH and designed to work similar to FTP, but over a secure connection.

SFTP allows you to limit the number of ports used for file transfer connections. It is preferable to FTP because of its underlying security features and ability to use a standard SSH connection.

- [Generating the Required SSH Keys](#)
To enable SFTP, you must generate SSH keys. This procedure needs to be done only once on one of the Managed Servers, because Managed File Transfer shares the same SFTP key for all the servers in the cluster.
- [Configuring the SFTP Ports](#)
Before you can use the Secure File Transfer Protocol (SFTP) for Oracle Managed File Transfer, you must configure the SFTP Ports.
- [Configuring Oracle Load Balancer for SFTP Services](#)
- [Additional SFTP Configuration Steps for Managed File Transfer](#)
There are several additional configuration steps that you should perform when you use SFTP with Managed File Transfer.

Generating the Required SSH Keys

To enable SFTP, you must generate SSH keys. This procedure needs to be done only once on one of the Managed Servers, because Managed File Transfer shares the same SFTP key for all the servers in the cluster.

Without a valid private key, SSH-FTP server fails to start. To comply with security best practices, you should always use a password-protected private key. Perform the following steps to create the SSH key and configure it for the SFTP embedded servers:

1. Log into `MFTHOST1` and run the `ssh-keygen` command to generate a key.

Example

```
ssh-keygen \-t rsa \-b 2048 -m pem
```

`ssh-keygen` is a standard Unix and Linux command. For more information, see your Operating System documentation.

- a. Enter the **location** for the file in which to save the key.
- b. Enter a **passphrase** to protect the key.

Make a note of this information as you need it later.

2. Import the key into the Managed File Transfer keystore:
 - a. Make sure that the Managed File Transfer Managed Servers are up and running.
 - b. Change directory to the following location:

```
cd $ORACLE_COMMON_HOME/common/bin
```


- c. Start the WebLogic Server Scripting Tool (WLST):

```
export WLST_PROPERTIES="-Dweblogic.security.TrustKeyStore=CustomTrust
-Dweblogic.security.CustomTrustKeyStoreFileName=/u01/oracle/config/
keystores/appTrustKeyStore.pkcs12
-Dweblogic.security.CustomTrustKeyStorePassPhrase=password"
./wlst.sh
```

- d. Connect to the Administration Port of the first Managed Server, by using the following command syntax:

```
connect('admin_user','admin_password','server_url')
```

Example

```
connect('weblogic','<password>','t3s://MFTHOST1:9014')
```

- e. Run the following WLST command to import the key:

```
importCSFKey('SSH', 'PRIVATE', 'alias', 'pvt_key_file_path')
```

Replace *alias* with the a name that you can use to identify the Managed Server.

Replace *pvt_key_file_path* with the name and directory location of the key that you generated earlier in this procedure. See *importCSFKey* in *WLST Command Reference for SOA Suite*

3. After you successfully import the SSH key, enable SSH-FTP and select the private key alias:

- a. Connect to the Managed File Transfer console at the following URL, by using the domain administration user and password:

```
https://mft.example.com/mftconsole
```

- b. Navigate to the **Administration** tab, and then **Keystore Management**.
- c. In the **SSH Keystore** section, enter the passphrase you used in *Step 1* to protect the key in the **Private Key Password** field.
- d. Save the changes that you just made.
- e. Select the **Administration** tab, and in the navigation tree, expand **Embedded Servers**.
- f. On the SFTP tab, select **Enabled**.
- g. Select the private key alias you created earlier in this procedure from the **Host Key Alias** drop-down menu.
- h. Leave the **Authentication Type** to **Password**.
- i. Save your changes.
- j. Click **Start** to start the SSH-FTP service.

Configuring the SFTP Ports

Before you can use the Secure File Transfer Protocol (SFTP) for Oracle Managed File Transfer, you must configure the SFTP Ports.

1. Connect to the Managed File Transfer console, by using the domain admin user name and password:

```
https://mft.example.com/mftconsole
```

2. Select the **Administration** tab.
3. In the left navigation pane, expand **Embedded Servers**.
4. Click **Ports**.
5. Enter `7022` as the **Configured Port** for the Managed File Transfer servers SFTP services.
6. Click **Save**.
7. Select all and click **Restart** to restart the server instances.

Configuring Oracle Load Balancer for SFTP Services

As described in the [Configuring Virtual Hosts on the Hardware Load Balancer](#), the Managed File Transfer requires a TCP virtual server in the load balancer for the Secure File Transfer Protocol (SFTP), in addition to the virtual server for HTTPS. This TCP virtual server directly routes the SFTP requests to the SFTP embedded servers that run on the Managed File Transfer Managed Servers. For consistency, the port used in the hardware load balancer and in the SFTP servers is the same (7022). Ensure you have created the appropriate resources (services, pools, and so on) in your load balancer for this virtual server to allow access to the SFTP through the load balancer.

Additional SFTP Configuration Steps for Managed File Transfer

There are several additional configuration steps that you should perform when you use SFTP with Managed File Transfer.

1. Connect to the Managed File Transfer console at the following URL:
`https://mft.example.com/mftconsole`
2. Select **Administration**, and then in the navigation tree, select **Embedded Servers**.
 - a. Update the **Root Directory** so that it points to a shared storage.

Example

```
ORACLE_RUNTIME/mftedg_domain/MFT_Cluster/ftp_root
```

- b. Click **Save**.
3. Select **Administration**, and then in the navigation tree, select **Server Properties**.
4. Update the High Availability Properties:
 - a. Update the payload and callout directories so that they point to a shared storage location that can be accessed by the different servers in the cluster.

Example

```
ORACLE_RUNTIME/mftedg_domain/MFT_Cluster/storage
```

```
ORACLE_RUNTIME/mftedg_domain/MFT_Cluster/callouts
```

- b. Set the **Control Directory** to a shared location.

Example

```
ORACLE_RUNTIME/mftedg_domain/MFT_Cluster/control_dir
```

The **Control Directory** is the directory path that the Managed File Transfer File and FTP adapters use to handle high availability use cases. This field is required if the MFT is running in HA environment. You must set it to a shared location if multiple Oracle WebLogic Server instances run in a cluster.

- c. Verify the values in the following fields.
 - Verify that the value for **Inbound Datasource** is set to `jdbc/MFTDataSource`.
 - Verify that the value for **Outbound Datasource** is set to `jdbc/MFTDataSource`.
- d. Save the changes that you made so far.
- e. In the Navigation tree, expand **Advanced Delivery Properties**.

The Advanced Delivery Properties capture the Internal Address and External Address (IP addresses) and the FTP, FTPS, and SFTP ports that the load balancer uses.

Use these settings when Oracle Managed File Transfer sends a payload as an FTP or SFTP reference. If the values are set, they are used to construct the FTP reference (FTP/SFTP host address and ports).

If Managed File Transfer is running behind internal and external proxies, then the Internal and External IP addresses are required.

- **Internal Address:** Leave this field blank, unless you use an internal load balancer for SFTP. The default enterprise deployment uses an external load balancer, but not an internal load balancer.
- **External Address:** Enter the address that is used as the entry point for your SFT requests through the external load balancer.

For example, enter `sftp.example.com` as the address and `7022` as SFTP port.

- f. Save the changes you made and exit the console.
5. Restart the WLS_MFT Managed servers.
6. Use any standard SFTP client application to verify that you can use SFTP to access the Managed File Transfer servers.

Example

```
sftp -o "Port 7022" weblogic@MFTHOST1
Connecting to MFTHOST1 ...
Password authentication
Password:
sftp>
```

7. Use any standard SFTP client application to verify that you can use SFTP to access the Managed File Transfer servers through the load balancer.

Example

```
sftp -o "Port 7022" weblogic@mft.example.com
Connecting to mft.example.com ...
Password authentication
Password:
sftp>
```

Creating a New LDAP Authenticator and Provisioning Users for Managed File Transfer

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.

This procedure is required for each new Oracle Fusion Middleware domain. For an Oracle Managed File Transfer domain, you can perform the following tasks:

1. Review [Creating a New LDAP Authenticator and Provisioning Enterprise Deployment Users and Group](#) to understand the required concepts and to create the new LDAP Authenticator.
2. When you provision the users and groups, use the following user and group names for Managed File Transfer administration authentication:
Administrative user: `weblogic_mft`
Administrative group: `MFT Administrators`
3. Assign product-specific administration role to the group by logging in to Oracle Enterprise Manager Fusion Middleware Control. See [Configuring Roles for Administration of an Enterprise Deployment](#).

Replacing Connect Strings with the Appropriate TNS Alias

Oracle recommends using TNS Alias in the connection strings used by FMW components instead of repeating long JDBC strings across multiple connections pools.

For more information about how to use TNS alias in your Datasources, see [Using TNS Alias in Connect Strings](#) in the *Common Configuration and Management Tasks for an Enterprise Deployment* chapter.

Backing Up the Configuration

It is an Oracle best practices recommendation to create a backup after you successfully extended a domain or at another logical point. Create a backup after you verify that the installation so far is successful. This is a quick backup for the express purpose of immediate restoration in case of problems in later steps.

The backup destination is the local disk. You can discard this backup when the enterprise deployment setup is complete. After the enterprise deployment setup is complete, you can initiate the regular deployment-specific Backup and Recovery process.

For information about backing up your configuration, see [Performing Backups and Recoveries for an Enterprise Deployment](#).

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 Service Migration in an Enterprise Deployment](#)
The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. Whole Server Migration requires more resources and a full start of a managed server, so it involves a higher failover latency than Service Migration. The products included in this EDG support Service Migration. Hence, Service Migration is recommended and this guide explains how to use Service Migration in an Oracle Fusion Middleware enterprise topology. Whole Server migration is out of the scope of this guide.
- [Scaling Procedures for an Enterprise Deployment](#)
The scaling procedures for an enterprise deployment include scale out, scale in, scale up, and scale down. During a scale-out operation, you add managed servers to new nodes. You can remove these managed servers by performing a scale in operation. During a scale-up operation, you add managed servers to existing hosts. You can remove these servers by performing a scale-down operation.

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.

- [Configuration and Management Tasks for All Enterprise Deployments](#)
These are some of the typical configuration and management tasks you are likely need to perform on an Oracle Fusion Middleware enterprise deployment.
- [Configuration and Management Tasks for an Oracle SOA Suite Enterprise Deployment](#)
These are some of the key configuration and management tasks that you likely need to perform on an Oracle SOA Suite enterprise deployment.

Configuration and Management Tasks for All Enterprise Deployments

These are some of the typical configuration and management tasks you are likely need to perform on an Oracle Fusion Middleware enterprise deployment.

- [Verifying Appropriate Sizing and Configuration for the WLSRuntimeSchemaDataSource](#)
In Oracle FMW 14.1.2, `WLSRuntimeSchemaDataSource` is the common datasource that is reserved for use by the FMW components for JMS JDBC Stores, JTA JDBC stores, and Leasing services. `WLSRuntimeSchemaDataSource` is used to avoid contention in critical WLS infrastructure services and to guard against dead-locks.
- [Verifying Manual Failover of the Administration Server](#)
In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and fallback of the Administration Server from SOAHOST1 and SOAHOST2 are detailed in the following sections.
- [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#)
After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur.
- [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 SOA Suite enterprise deployment, Oracle recommends that you perform this task after you verify the SOA Managed Servers.
- [About Using Third Party SSL Certificates in the WebLogic and Oracle HTTP Servers](#)
This Oracle SOA Suite Enterprise Deployment Topology uses SSL all the way from the external clients to the backend WebLogic Servers. The previous chapters in this guide provided scripts (`generate_perdomainCACERTS.sh` and `generate_perdomainCACERTS-ohs.sh`) to generate the required SSL certificates for the different FMW components.

- [Enabling SSL Communication Between the Middle Tier and SSL Endpoints](#)
It is important to understand how to enable SSL communication between the middle tier and the frontend hardware load balancer or any other external SSL endpoints that needs to be accessed by the SOA Suite WebLogic Server. For example, for external webservices invocations, callbacks, and so on.
- [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.
- [Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment](#)
The Oracle WebLogic persistent store framework 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.
- [Best Configuration Practices When Using RAC and Gridlink Datasources](#)
Oracle recommends that you use GridLink data sources when you use an Oracle RAC database. If you follow the steps described in the Enterprise Deployment guide, the datasources will be configured as GridLink.
- [Using TNS Alias in Connect Strings](#)
You can create an alias to map the URL information instead of specifying long database connection strings in the jdbc connection pool of a datasource. The connection string information is stored in a `tnsnames.ora` file with an associated alias name. This alias is used in the connect string of the connection pool.
- [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 SOA Suite enterprise deployment.

Verifying Appropriate Sizing and Configuration for the WLSRuntimeSchemaDataSource

In Oracle FMW 14.1.2, `WLSRuntimeSchemaDataSource` is the common datasource that is reserved for use by the FMW components for JMS JDBC Stores, JTA JDBC stores, and Leasing services. `WLSRuntimeSchemaDataSource` is used to avoid contention in critical WLS infrastructure services and to guard against dead-locks.

To reduce the `WLSRuntimeSchemaDataSource` connection usage, you can change the JMS JDBC and TLOG JDBC stores connection caching policy from *Default* to *Minimal* by using the respective connection caching policy settings. When there is a need to reduce connections in the back-end database system, Oracle recommends that you set the caching policy to *Minimal*. Avoid using the caching policy *None* because it causes a potential degradation in performance. For a detailed tuning advice about connections that are used by JDBC stores, see *Configuring a JDBC Store Connection Caching Policy* in *Administering the WebLogic Persistent Store*.

The default `WLSRuntimeSchemaDataSource` connection pool size is 75 (size is double in the case of a GridLink DataSource). You can tune this size to a higher value depending on the size of the different FMW clusters and the candidates that are configured for migration. For example, consider a typical SOA EDG deployment with the default number of worker threads per store. If more than 25 JDBC Stores or TLOG-in-DB instances or both can fail over to the

same WebLogic server, and the Connection Caching Policy is not changed from *Default* to *Minimal*, possible connection contention issues could arise. In these cases, increasing the default `WLSRuntimeSchemaDataSource` pool size (maximum capacity) becomes necessary (each JMS store uses a minimum of two connections, and leasing and JTA are also added to compete for the pool).

Verifying Manual Failover of the Administration Server

In case a host computer fails, you can fail over the Administration Server to another host. The steps to verify the failover and failback of the Administration Server from SOAHOST1 and SOAHOST2 are detailed in the following sections.

Assumptions:

- The Administration Server is configured to listen on ADMINVHN or any custom virtual host that maps to a floating IP/VIP. It should not listen on ANY (blank listen address), localhost or any host name that uniquely identifies a single node.

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 SOAHOST1 to SOAHOST2, and the two nodes have these IPs:
 - SOAHOST1: 100.200.140.165
 - SOAHOST2: 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 SOAHOST1 or SOAHOST2.
- Oracle WebLogic Server and Oracle Fusion Middleware components have been installed in SOAHOST2 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 When Using a Per Host Node Manager](#)
The following procedure shows how to fail over the Administration Server to a different node (SOAHOST2). 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 SOAHOST2 Through Load Balancer](#)
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.
- [Failing the Administration Server Back to SOAHOST1 When Using a Per Host Node Manager](#)
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 When Using a Per Host Node Manager

The following procedure shows how to fail over the Administration Server to a different node (SOAHOST2). 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 host Node Manager for the enterprise topology, as described in [Creating a Per Host Node Manager Configuration](#). For more information, see [About the Node Manager Configuration in a Typical Enterprise Deployment](#).

To fail over the Administration Server to a different host:

1. Stop the Administration Server on SOAHOST1.
2. Stop the Node Manager on SOAHOST1.

You can use the script `stopNodeManager.sh` that was created in `NM_HOME`.

3. Migrate the ADMINVHN virtual IP address to the second host:
 - a. Run the following command as root on SOAHOST1 (where X is the current interface used by ADMINVHN) to check the virtual IP address at its CIDR:

```
ip addr show dev ethX
```

For example:

```
ip addr show dev eth0
```

- b. Run the following command as root on SOAHOST1 (where X is the current interface used by ADMINVHN):

```
ip addr del ADMINVHN/CIDR dev ethX
```

For example:

```
ip addr del 100.200.140.206/24 dev eth0
```

- c. Run the following command as root on SOAHOST2:

```
ip addr add ADMINVHN/CIDR dev ethX label ethX:Y
```

For example:

```
ip addr add 100.200.140.206/24 dev eth0 label eth0:1
```

 **Note:**

Ensure that the CIDR and interface to be used match the available network configuration in SOAHOST2.

4. Update the routing tables by using `arping`, for example:

```
arping -b -A -c 3 -I eth0 100.200.140.206
```

5. From SOAHOST1, change directory to the Node Manager home directory:

```
cd $NM_HOME
```

6. Edit the `nodemanager.domains` file and remove the reference to `ASERVER_HOME`.

The resulting entry in the SOAHOST1 `nodemanager.domains` file should appear as follows:

```
soaedg_domain=MSERVER_HOME;
```

7. From SOAHOST2, change directory to the Node Manager home directory:

```
cd $NM_HOME
```

8. Edit the `nodemanager.domains` file and add the reference to `ASERVER_HOME`.

The resulting entry in the SOAHOST2 `nodemanager.domains` file should appear as follows:

```
soaedg_domain=MSERVER_HOME;ASERVER_HOME
```

9. Start the Node Manager on SOAHOST1 and restart the Node Manager on SOAHOST2.
10. Start the Administration Server on SOAHOST2.
11. Check that you can access the Administration Server on SOAHOST2 and verify the status of components in Fusion Middleware Control using the following URL:

```
https://ADMINVHN:9002/em
```

Validating Access to the Administration Server on SOAHOST2 Through Load Balancer

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 SOAHOST2:

- `https://admin.example.com:445/em`

Where, 445 is the port you use to access to the Fusion Middleware Control in the Load Balancer.

This URL should display Oracle Enterprise Manager Fusion Middleware Control.

- Verify that you can log into the WebLogic Remote Console through the provider you defined for this domain.

Failing the Administration Server Back to SOAHOST1 When Using a Per Host Node Manager

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.

This procedure assumes that you have configured a per host Node Manager for the enterprise topology, as described in [Creating a Per Host Node Manager Configuration](#). For more information, see [About the Node Manager Configuration in a Typical Enterprise Deployment](#).

1. Stop the Administration Server on SOAHOST2.
2. Stop the Node Manager on SOAHOST2.
3. Run the following command as root on SOAHOST2.

```
ip addr del ADMINVHN/CIDR dev ethX
```

For example:

```
ip addr del 100.200.140.206/24 dev eth0
```

4. Run the following command as root on SOAHOST1:

```
ip addr add ADMINVHN/CIDR dev ethX label ethX:Y
```

For example:

```
ip addr add 100.200.140.206/24 dev eth0 label eth0:1
```

 **Note:**

Ensure that the CIDR and interface to be used match the available network configuration in SOAHOST1.

5. Update the routing tables by using `arping` on SOAHOST1:

```
arping -b -A -c 3 -I eth0 100.200.140.206
```

6. From SOAHOST2, change directory to the Node Manager home directory:

```
cd $NM_HOME
```

7. Edit the `nodemanager.domains` file and remove the reference to `ASERVER_HOME`.

8. From SOAHOST1, change directory to the Node Manager home directory:

```
cd $NM_HOME
```

9. Edit the `nodemanager.domains` file and add the reference to `ASERVER_HOME`.

10. Start the Node Manager on SOAHOST2 and restart the Node Manager on SOAHOST1.

11. Start the Administration Server on SOAHOST1.

12. Test that you can use the WebLogic Remote Console to access the provider defined for this domain.

13. Check that you can access and verify the status of components in the Oracle Enterprise Manager by using the following URL:

```
https://ADMINVHN:9002/em
```

```
https://admin.example.com:445/em
```

Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment

After you configure the domain and unpack it to the Managed Server domain directories on all the hosts, verify and update the upload and stage directories for Managed Servers in the new clusters. Also, update the upload directory for the AdminServer to have the same absolute path instead of relative, otherwise deployment issues can occur.

This step is necessary to avoid potential issues when you perform remote deployments and for deployments that require the stage mode.

To update the directory paths for the Deployment Stage and Upload locations, complete the following steps:

1. Log into the WebLogic Remote Console to access the provider of this domain.
2. Open the **Edit Tree**.
3. Expand **Environment**.

4. Expand **Servers**.
5. Click the name of the Managed Server you want to edit. Perform the following steps for each of the Managed Server:
 - a. Click the **Advanced** tab.
 - b. Click the **Deployment** tab.
 - c. Verify that the Staging Directory Name is set to the following:
`MSERVER_HOME/servers/server_name/stage`
Replace `MSERVER_HOME` with the full path for the `MSERVER_HOME` directory.
Update with the correct name of the Managed Server that you are editing.
 - d. Update the Upload Directory Name to the following value:
`ASERVER_HOME/servers/AdminServer/upload`
Replace `ASERVER_HOME` with the directory path for the `ASERVER_HOME` directory.
 - e. Click **Save**.
 - f. Return to the **Summary of Servers** screen.Repeat the same steps for each of the new managed servers.
6. Navigate to and update the Upload Directory Name value for the **AdminServer**:
 - a. Navigate to **Servers** and select the **AdminServer**.
 - b. Click the **Advanced** tab.
 - c. Click the **Deployment** tab
 - d. Verify that the Staging Directory Name is set to the following absolute path:
`ASERVER_HOME/servers/AdminServer/stage`
 - e. Update the Upload Directory Name to the following absolute path:
`ASERVER_HOME/servers/AdminServer/upload`
Replace `ASERVER_HOME` with the directory path for the `ASERVER_HOME` directory.
 - f. Click **Save**.
7. When you have modified all the appropriate objects, commit the changes in the shopping cart.
8. Restart all the Servers for the changes to take effect. If you are following the EDG steps in-order and are not going to make any deployments immediately, you can wait until the next restart.

 **Note:**

If you continue directly with further domain configurations, a restart to enable the stage and upload directory changes is not strictly necessary at this time.

Setting the Front End Host and Port for a WebLogic Cluster

You must set the front-end HTTP host and port for the Oracle WebLogic Server cluster that hosts the Oracle 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 SOA Suite 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 Remote Console:

1. Log in to the WebLogic Remote Console.
2. Open the **Edit Tree**.
3. Expand **Environment**.
4. Expand **Clusters**. On the **Clusters** page, click the cluster that you want to modify and then select the **HTTP** tab.
5. Use the information in [Table 19-1](#) to add the required frontend hostname and port to each cluster.

Table 19-1 The Frontend Hostname and Port for Each Cluster

Name	Cluster Address	Frontend Host	Frontend HTTP Port	Frontend HTTPS Port
SOA_Cluster	Leave it empty	soa.example.com	0	443
WSM-PM_Cluster	Leave it empty	soainternal.example.com	0	444
OSB_Cluster	Leave it empty	osb.example.com	0	443
ESS_Cluster	Leave it empty	soa.example.com	0	443
BAM_Cluster	Leave it empty	soa.example.com	0	443
MFT_Cluster	Leave it empty	mft.example.com	0	443

6. Click **Save**.
7. Commit the changes in the shopping cart.
8. Restart the managed servers of the cluster.

About Using Third Party SSL Certificates in the WebLogic and Oracle HTTP Servers

This Oracle SOA Suite Enterprise Deployment Topology uses SSL all the way from the external clients to the backend WebLogic Servers. The previous chapters in this guide provided scripts (`generate_perdomainCACERTS.sh` and `generate_perdomainCACERTS-ohs.sh`) to generate the required SSL certificates for the different FMW components.

These scripts generate the different SSL certificates using the WebLogic per domain Certification Authority in the WebLogic domain. These scripts also add the frontend's SSL certificates to the trust keystore. However, in a production environment, you may want to use your own SSL certificates, issued by your own or by a 3rd party certificate authority. This section provides you some guidelines to configure the EDG system with this type of SSL certificates.

- [Using Third Party SSL Certificates in WebLogic Servers](#)
- [Using Third Party SSL Certificates in Oracle HTTP Servers](#)

Using Third Party SSL Certificates in WebLogic Servers

Here are some guidelines about using custom or third party SSL certificates with the WebLogic Servers:

- The SSL certificate used by each WebLogic server (identity key, private key) must be issued to that server's listen address. For example, if the server *WLS_PROD1* listens in *apphost1.example.com*, the CN of its SSL certificate must be that hostname or wildcard name valid for that hostname.
- Oracle recommends using an identity keystore shared by all the servers in the same domain where you import all the private keys used by the different WebLogic servers each mapped to a different alias.
- Oracle recommends using a trust keystore shared by all the servers in the domain. You must import the Certificate Authority's certificate (and intermediate and root CA if needed) into this trust keystore.
- You must specify the identity keystore, alias of the identity key and the trust keystore for each WebLogic server in the WebLogic domain's configuration. Use WebLogic's Remote Console to configure these SSL settings for each server.
- Start the WebLogic servers using the appropriate java options to point to the trusted keystore so that they can communicate with external SSL endpoints that use the Certificate Authorities included in such a trust store.

The following commands are useful to manage SSL certificates in WebLogic.

- Command to import an SSL certificate (a private key) into the identity keystore:

Syntax

```
WL_HOME/server/bin/setWLSEnv.sh
```

```
java utils.ImportPrivateKey
  -certfile cert_file
  -keyfile private_key_file
  [-keyfilepass private_key_password]
  -keystore keystore
  -storepass storepass
  [-storetype storetype]
  -alias alias
  [-keypass keypass]
```

Example for a Certificate Issued to *apphost1.example.com*

```
WL_HOME/server/bin/setWLSEnv.sh
```

```
java utils.ImportPrivateKey \
  -certfile apphost1.example.com_cert.der \
  -keyfile apphost1.example.com_key.der \
  -keyfilepass keypassword \
  -storetype pkcs12 \
  -keystore CustomIdentityKeystore.pkcs12 \
  -storepass keystorepassword \
  -alias apphost1.example.com \
  -keypass keypassword
```

- Command to import an SSL certificate (a trusted certificate) into the trusted keystore:

Syntax

```
keytool -import -v -noprompt -trustcacerts \
  -alias <alias_for_trusted_cert> \
  -file <certificate>.der \
  -storetype <keystoretype> \
  -keystore <customTrustKeyStore> \
  -storepass <keystorepassword>
```

Example for Importing a CA Certificate

```
keytool -import -v -noprompt -trustcacerts \
  -alias example_ca_cert \
  -file example_ca_cert.der \
  -storetype pkcs12 \
  -keystore CustomTrustKeyStore.pkcs12 \
  -storepass keystorepassword
```

Example of the Java Options for Servers to Load Custom Trust Keystore

```
EXTRA_JAVA_PROPERTIES="{EXTRA_JAVA_PROPERTIES}
  -Djavax.net.ssl.trustStore=/u01/oracle/config/keystores/
CustomTrustKeyStore.pkcs12
  -Djavax.net.ssl.trustStorePassword=<keystorepassword>"
export EXTRA_JAVA_PROPERTIES
```

Using Third Party SSL Certificates in Oracle HTTP Servers

Here are some guidelines to use your own SSL certificates in OHS:

- Each OHS virtual host using SSL must use a wallet that contains only one private key. This private key will be used as the OHS server's SSL certificate. It must be issued to the hostname in which the virtual host listens (the hostname value in the "VirtualHost" directive). The private key can also include other hostnames such as Subject Alternative Name (SAN) names (for example, the value of the "ServerName" directive). The virtual host must include the **SSLWallet** directive pointing to this wallet.
- Different OHS virtual hosts can use the same **SSLWallet** (hence, the same private key), as long as they use the same hostname in the VirtualHost directive. The port can be different.
- OHS acts as a client when it connects to the WebLogic servers. Hence, it must trust the certificate authority that issued the WebLogic's certificates. Use the directive **WLSSLWallet** in the `mod_wl_ohs.conf` file to point to the appropriate wallet that contains the WebLogic certificates' CA cert.
- The frontend load balancer acts as a client when it connects to the OHS servers. It must trust the certificate authority that issued the certificates used by OHS. You must check your load balancer documentation to import the OHS's CA as a trusted authority.

The following commands are useful to manage keys and wallets in OHS.

- Command to create a wallet for OHS (orapki):

Syntax

```
$WEB_ORACLE_HOME/bin/orapki wallet create \  

-wallet wallet \  

-auto_login_only
```

Example

```
$WEB_ORACLE_HOME/bin/orapki wallet create \  

-wallet /u02/oracle/config/keystores/orapki/ \  

-auto_login_only
```

- Command to add a private key to a wallet (orapki) from an identity keystore:

Syntax

```
$WEB_ORACLE_HOME/bin/orapki wallet jks_to_pkcs12 \  

-wallet wallet \  

-pwd pwd \  

-keystore keystore \  

-jkspwd keystorepassword  

[-aliases [alias:alias..]]
```

Example

```
$WEB_ORACLE_HOME/bin/orapki wallet jks_to_pkcs12 \  

-wallet /u02/oracle/config/keystores/orapki/ \  

-keystore /u02/oracle/config/keystores/customIdentityKeyStore.pkcs12 \  

-jkspwd keystorepassword \  

-aliases ohshost1.example.com
```

- Command to add all the trusted keys to a wallet (orapki) from a trusted keystore:

Example

```
$WEB_ORACLE_HOME/bin/orapki wallet jks_to_pkcs12 \  

-wallet /u02/oracle/config/keystores/orapki/ \  

-keystore /u02/oracle/config/keystores/customTrustKeyStore.pkcs12 \  

-jkspwd password
```

- Command to list all the keys of a wallet (orapki):

Example

```
$WEB_ORACLE_HOME/bin/orapki wallet display \  

-wallet /u02/oracle/config/keystores/orapki/
```


Enabling SSL Communication Between the Middle Tier and SSL Endpoints

It is important to understand how to enable SSL communication between the middle tier and the frontend hardware load balancer or any other external SSL endpoints that needs to be accessed by the SOA Suite WebLogic Server. For example, for external webservice invocations, callbacks, and so on.

Note:

The following steps are applicable if the hardware load balancer is configured with SSL and the frontend address of the system has been secured accordingly.

- [When is SSL Communication Between the Middle Tier and the Frontend Load Balancer Necessary?](#)
- [Generating Certificates, Identity Store, and Truststores](#)
- [Importing Other External Certificates into the Truststore](#)
- [Adding the Updated Trust Store to the Oracle WebLogic Server Start Scripts](#)

When is SSL Communication Between the Middle Tier and the Frontend Load Balancer Necessary?

In an enterprise deployment, there are scenarios where the software running on the middle tier must access the frontend 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.

For example, the following examples are applicable in an Oracle SOA Suite enterprise deployment:

- Oracle Business Process Management and SOA Composer require access to the frontend load balancer URL when they attempt to retrieve role and security information through specific web instances. Some of these invocations require not only that the LBR certificate is added to the WebLogic Server's trust store but also that the appropriate identity key certificates are created for the SOA server's listen addresses.
- Oracle Service Bus performs invocations to endpoints exposed in the Load Balancer SSL virtual servers.
- Oracle SOA Suite composite applications and services often generate callbacks that need to perform invocations by using the SSL address exposed in the load balancer.
- Oracle SOA Suite composite applications and services often access external webservices using SSL.
- Finally, when you test a SOA Web services endpoint in Oracle Enterprise Manager Fusion Middleware Control, the Fusion Middleware Control software that is running on the Administration Server must access the load balancer frontend to validate the endpoint.

Generating Certificates, Identity Store, and Truststores

Since this Enterprise Deployment Guide uses end to end SSL (except in the access to the Database), certificates have already been generated in the different chapters using a per-domain CA. These have been already added to the pertaining Identity Stores and a Truststore has also been configured to include the per-domain CA. It is expected that through the use of the different generateCerts scripts provided, appropriate certificates exist already in these stores for the different listen addresses used by the WebLogic servers in the domain. On top of this, when the script `generate_perdomainCACERTS-ohs.sh` is executed, it traverses all the front-end addresses in the domain's `config.xml` and adds its pertaining certificates to the trust store used by the domain. By adding these trust stores to the java properties used by the WebLogic Servers in the domain (`-Djavax.net.ssl.trustStore` and `-Djavax.net.ssl.trustStorePassword`), the appropriate SSL handshake is guaranteed when these WebLogic servers acts as client sin SSL invocations.

Importing Other External Certificates into the Truststore

Perform the following steps to add any other SSL end point's certificates to the domain's truststore. These may be external addresses or frontends in other WLS domains used by the applications in the SOA EDG one:

1. Access the end point's site on SSL with a browser (this adds the server's certificate to the browser's repository).
2. Obtain the certificate from the site. For example, you can obtain a webservice site's certificate using a browser such as Firefox. From the browser's certificate management tool, export the certificate to a file that is on the server's file system (with a file name such as `site.webservice.com.crt`). Alternatively, you can obtain the certificate using the `openssl` command. The syntax of the commands is as follows:

```
openssl s_client -connect site.webservice.com -showcerts </dev/null 2>/dev/null|openssl x509 -outform PEM > $KEYSTORE_HOME/ site.webservice.com.crt
```

3. Use the `keytool` to import the site's certificate into the truststore:

For example:

```
keytool -import -file /oracle/certificates/site.webservice.com.crt -v -keystore appTrustKeyStore.pkcs12 -alias siteWS -storepass password
```

4. Repeat this procedure for each SSL endpoint accessed by your WebLogic Servers.

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

Since the trust store's path was already added to the WebLogic start scripts in the chapter where the domain was created, no additional configuration is required. Simply ensure that the new trust store (with the CAs and/or certs for the SSL endpoints added) replaces the existing one.

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.

- [Summary of Products with Specific Administration Roles](#)
- [Summary of Oracle SOA Suite Products with Specific Administration Groups](#)
- [Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group](#)
- [Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group](#)

Summary of Products with Specific Administration Roles

The following table lists the Fusion Middleware products that have specific administration roles, which must be added to the enterprise deployment administration group (SOA Administrators), which you defined in the LDAP Authorization Provider for the enterprise deployment.

Use the information in the following table and the instructions in [Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group](#) to add the required administration roles to the enterprise deployment Administration group.

Product	Application Stripe	Administration Role to be Assigned
Oracle Web Services Manager	wsm-pm	policy.updater
SOA Infrastructure	soa-infra	SOAAdmin
Oracle Service Bus	Service_Bus_Console	MiddlewareAdministrator

Product	Application Stripe	Administration Role to be Assigned
Enterprise Scheduler Service	ESSAPP	ESSAdmin
Oracle B2B	b2bui	B2BAdmin
Oracle MFT	mftapp	MFTAdmin
Oracle MFT	mftes	MFTESAdmin

Summary of Oracle SOA Suite Products with Specific Administration Groups

Table 19-2 lists the Oracle SOA Suite products that need to use specific administration groups.

For each of these components, the common enterprise deployment Administration user must be added to the product-specific Administration group; otherwise, you won't be able to manage the product resources by using the enterprise manager administration user that you created in [Provisioning an Enterprise Deployment Administration User and Group](#).

Use the information in Table 19-2 and the instructions in [Adding the Enterprise Deployment Administration User to a Product-Specific Administration Group](#) to add the required administration roles to the enterprise deployment Administration group.

Table 19-2 Oracle SOA Suite Products with a Product-Specific Administration Group

Product	Product-Specific Administration Group
Oracle Business Activity Monitoring	BAMAdministrator
Oracle Business Process Management	Administrators
Oracle Service Bus Integration	IntegrationAdministrators
MFT	OracleSystemGroup

Note:

MFT requires a specific user, namely OracleSystemUser, to be added to the central LDAP. This user must belong to the OracleSystemGroup group. You must add both the user name and the user group to the central LDAP to ensure that MFT job creation and deletion work properly.




Adding a Product-Specific Administration Role to the Enterprise Deployment Administration Group

For products that require a product-specific administration role, use the following procedure to add the role to the enterprise deployment administration group:

1. Sign-in to the Fusion Middleware Control by using the administrator's account (for example: `weblogic_soa`), and navigate to the home page for your application.

These are the credentials that you created when you initially configured the domain and created the Oracle WebLogic Server Administration user name (typically, `weblogic_soa`) and password.

2. From the **WebLogic Domain** menu, select **Security**, and then **Application Roles**.

3. For each production-specific application role, select the corresponding application stripe from the **Application Stripe** drop-down menu.
4. Click Search Application Roles icon  to display all the application roles available in the domain.
5. Select the row for the application role that you are adding to the enterprise deployment administration group.
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 (for example, SOA Administrators) in the **Principal Name Starts With** field and clicking the right arrow to start the search.
10. Select the administrator group in the search results and click **OK**.
11. Click **OK** on the Edit Application Role page.

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_soa`) 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_soa, cn=users, dc=us, dc=oracle, dc=com
cn: product-specific_group_name
description: Administrators Group for the Domain
```

In this example, replace `product-specific_group_name` with the actual name of the product administrator group, as shown in [Table 19-2](#).

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 Remote Console.

2. Use the **ldif** file to add the enterprise deployment administrator user to the product-specific administration group.

For Oracle Unified Directory:

```
OID_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
```

Using Persistent Stores for TLOGs and JMS in an Enterprise Deployment

The Oracle WebLogic persistent store framework 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. It needs to be noted that the configuration wizard steps provided in the different chapters in this book will already create JDBC persistent stores for the components used. Use the manual steps below for custom stores or for transitioning to JDBC stores from file stores.

- [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](#)

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
MFT	Yes	Yes
OSB	Yes	Yes
SOA	Yes	Yes
WSM	No	No

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 identify 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. For example, the impact of switching from a file-based to

database-based persistent store is very low when you use the SOA Fusion Order Demo (a sample application used to test Oracle SOA Suite environments), because the JMS database operations are masked by many other SOA database invocations that are much heavier.

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 lob)
- 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 asynchronous, 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 Partitioning

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.

Note:

Remember that the steps provided for setting up the different components in this EDG (using the configuration wizard) is already configured in JDBC persistent stores for them. Use the following steps for custom persistent stores or when reconfiguring from file stores to JDBC stores (migration of messages from file to JDBC is out of the scope of this EDG).

- [Recommendations for TLOGs and JMS Datasource Consolidation](#)
To accomplish data source consolidation and connection usage reduction, use a single connection pool for both JMS and TLOGs persistent stores.
- [Roadmap for Configuring a JDBC Persistent Store for TLOGs](#)
The following topics describe how to configure a database-based persistent store for transaction logs.
- [Roadmap for Configuring a JDBC Persistent Store for JMS](#)
The following topics describe how to configure a database-based persistent store for JMS.
- [Creating a User and Tablespace for TLOGs](#)
Before you can create a database-based persistent store for transaction logs, you must create a user and tablespace in a supported database.
- [Creating a User and Tablespace for JMS](#)
Before you can create a database-based persistent store for JMS, you must create a user and tablespace in a supported database.
- [Creating GridLink Data Sources for TLOGs and JMS Stores](#)
Before you can configure database-based persistent stores for JMS and TLOGs, you must create two data sources: one for the TLOGs persistent store and one for the JMS persistent store.
- [Assigning the TLOGs JDBC Store to the Managed Servers](#)
If you are going to accomplish data source consolidation, you will reuse the `<PREFIX>_WLS` tablespace and `WLSRuntimeSchemaDataSource` for the TLOG persistent store. Otherwise, ensure that you create the tablespace and user in the database, and you have created the datasource before you assign the TLOG store to each of the required Managed Servers.
- [Creating a JDBC JMS Store](#)
After you create the JMS persistent store user and table space in the database, and after you create the data source for the JMS persistent store, you can then use the WebLogic Remote 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 `WLSRuntimeSchemaDataSource` as is for TLOGs and JMS persistent stores under non-high workloads and consider increasing the `WLSRuntimeSchemaDataSource` 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 `WLSRuntimeSchemaDataSource` 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 `WLSRuntimeSchemaDataSource` 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. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Services** and select **Data Sources**.
4. In the Summary of **Data Sources** page, click **New** and select **GridLink Data Source**. Enter the following:

Table 19-3 GridLink Data Source Properties

Properties	Description
Name	Enter a logical name for the data source in the Name field. For example, Leasing.
JNDI Names	Enter a name for JNDI. For example, for the TLOGs store enter jdbc/tlogs. For the JMS store, enter jdbc/jms.
Targets	Select the cluster that is using the persistent store and move to "Chosen".
Data Source Type	Select GridLink Data Source .
Database Driver	Select Oracle's Driver (Thin) for GridLink Connections Versions: Any .
Global Transaction Protocol	Select None .
Listeners	Enter the SCAN address and port for the RAC database, separated by a colon. For example, db-scan.example.com:1521.
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, soaedg.example.com.
Database username	Enter the user name. For example, 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.
Protocol	Leave the default value (TCP).
Fan Enabled	This property must be checked.
ONS Nodes	You can leave this field empty. ONS node list is automatically retrieved when the database is 12.2 or higher version.
ONS Wallet and password	You can leave this field empty.
Test Configuration	You must enable this option.

5. Click **Create**.
6. Commit changes in the shopping cart.
7. Repeat *Step 4* to *Step 6* to create the GridLink Data Source for JMS File Stores.

Assigning the TLOGs JDBC Store to the Managed Servers

If you are going to accomplish data source consolidation, you will reuse the `<PREFIX>_WLS` tablespace and `WLSRuntimeSchemaDataSource` for the TLOG persistent store. Otherwise, ensure that you create the tablespace and user in the database, and you have created the datasource before you assign the TLOG store to each of the required Managed Servers.

1. Log into the Oracle WebLogic Remote Console.
2. In the **Edit Tree**, navigate to **Environment > Servers**.
3. Click the name of the Managed Server.
4. Select the **Services > JTA** tab.
5. Enable **Transaction Log Store in JDBC**.
6. In the **Data Source** menu, select **WLSSchemaRuntimeDatasource** to accomplish data source consolidation. The `<PREFIX>_WLS` tablespace will be used for TLOGs.
7. In the **Transaction Log Prefix Name** field, specify a prefix name to form a unique **JDBC TLOG** store name for each configured JDBC TLOG store.
8. Click **Save**.
9. Repeat *step 2* to *step 7* for each additional managed server.
10. To activate these changes, commit the changes in the shopping cart.

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 WebLogic Remote Console to create the store.

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Services** and select **JDBC Stores**.
4. Click **New**.
5. Enter a persistent store name that easily relates it to the pertaining JMS servers that is using it.

Note:

To accomplish data source consolidation, select `WLSRuntimeSchemaDataSource`. The `<PREFIX>_WLS` tablespace is used for JMS persistent stores.

6. Target the store to the migratable target to which the JMS server belongs.
7. Repeat *Step 3* to *Step 7* for each additional JMS server in the cluster.
8. Commit changes in the shopping cart.

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 into the WebLogic Remote Console.
2. Navigate to the Edit Tree.
3. In the structure tree, expand **Services > Messaging > JMS Servers**.
4. Click the name of the JMS Server that you want to use the persistent store.
5. In the **Persistent Store** property, select the JMS persistent store you created.
6. Click **Save**.
7. Repeat *Step 3 to Step 6* for each of the additional JMS Servers in the cluster.
8. To activate these changes, commit changes in the shopping cart.

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 Remote Console to edit the existing JDBC Store you created earlier; create the table that is used for the JMS data:
 - a. Log into the WebLogic Remote Console.
 - b. Navigate to the **Edit Tree**.
 - c. In the structure tree, expand **Services** and select **JDBC stores**.
 - d. Click the persistent store you created earlier.
 - e. Click **Show Advanced Fields**.
 - f. Under the **Advanced** options, enter `ORACLE_RUNTIME/domain_name/ddl/jms_custom.ddl` in the **Create Table from DDL File** field.
 - g. Click **Save**.
 - h. To activate these changes, commit changes in the shopping cart.
5. Restart the Managed Servers.

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*.

Best Configuration Practices When Using RAC and Gridlink Datasources

Oracle recommends that you use GridLink data sources when you use an Oracle RAC database. If you follow the steps described in the Enterprise Deployment guide, the datasources will be configured as GridLink.

GridLink datasources provide dynamic load balancing and failover across the nodes in an Oracle Database cluster, and also receive notifications from the RAC cluster when nodes are added or removed. For more information about GridLink datasources, see *Using Active GridLink Data Sources in Administering JDBC Data Sources for Oracle WebLogic Server*.

Here is a summary of the best practices when using GridLink to connect to the RAC database:

- Use a database service (defined with `srvctl`) different from the default database service. In order to receive and process notifications from the RAC database, the GridLink needs to connect to a database service (defined with `srvctl`) instead to a default database service. These services monitor the status of resources in the database cluster and generate

notifications when the status changes. A database service is used in Enterprise Deployment guide, created and configured as described in [Creating Database Services](#).

- Use the long format database connect string in the datasources
When Gridlink datasources are used, the long format database connect string must be used. The Configuration Wizard does not set the long format string, it sets the short format instead. You can modify it manually later to set the long format. To update the datasources:

1. Connect to the WebLogic Server Console and navigate to **Domain Structure > Services > Datasources**.
2. Select a datasource, click the **Configuration** tab, and then click the **Connection Pool** tab.

3. Within the JDBC URL, change the **URL** from `jdbc:oracle:thin:[SCAN_VIP]:[SCAN_PORT]/[SERVICE_NAME]` to
`jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=[SCAN_VIP])(PORT=[SCAN_PORT])))`
`(CONNECT_DATA=(SERVICE_NAME=[SERVICE_NAME])))`

For example:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(LOAD_BALANCE=ON)
(ADDRESS=(PROTOCOL=TCP)(HOST=db-scan-address)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=soaedg.example.com)))
```

- Use auto-ons
The ONS connection list is automatically provided from the database to the driver. You can leave the ONS Nodes list empty in the datasources configuration.
- Test Connections On Reserve
Verify that the **Test Connections On Reserve** is checked in the datasources.

Eventhough the GridLink datasources receive FAN events when a RAC instances becomes unavailable, it is a best practice to enable the **Test Connections On Reserve** in the datasource and ensure that the connection returned to the application is good.
- Seconds to Trust an Idle Pool Connection
For a maximum efficiency of the test, you can also set **Seconds to Trust an Idle Pool Connection** to 0, so the connections are always verified. Setting this value to zero means that all the connections returned to the application will be tested. If this parameter is set to 10, the result of the previous test will be valid for 10 seconds and if a connection is reused before the lapse of 10 seconds, the result will still be valid.
- Test Frequency
Verify that the **Test Frequency** parameter value in the datasources is not 0. This is the number of seconds a WebLogic Server instance waits between attempts when testing unused connections. The default value of 120 is normally enough.

Using TNS Alias in Connect Strings

You can create an alias to map the URL information instead of specifying long database connection strings in the jdbc connection pool of a datasource. The connection string information is stored in a `tnsnames.ora` file with an associated alias name. This alias is used in the connect string of the connection pool.

The following example is of a connect string using tns alias.

```
jdbc:oracle:thin:@soaedg_alias
```


The `tnsnames.ora` file contains the following details.

```
soaedg_alias =
  (DESCRIPTION=
    (ADDRESS_LIST=
      (LOAD_BALANCE=ON)
      (ADDRESS=(PROTOCOL=TCP) (HOST=soaedgdb-scan) (PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME=soaedg.example.com))
  )
```

You must specify the `oracle.net.tns_admin` property in the datasource configuration to point to a specific `tnsnames.ora` file. For example, `<property><name>oracle.net.tns_admin</name><value>/u01/oracle/config/domains/fmw1412edg/config/tnsadmin</value></property></properties>`

This is the Maximum Availability and Enterprise Deployment recommended approach for JDBC urls. It simplifies JDBC configurations, facilitates DB configuration aliasing in disaster protection scenarios, and makes database connection changes more dynamic. For more information, see [Use a TNS Alias Instead of a DB Connection String in Administering JDBC Data Sources for Oracle WebLogic Server](#).

In Oracle Fusion Middleware 14.1.2, you can use a new type of deployment module to manage the `tnsnames.ora` files, wallet files, and keystore and truststore files associated with a database connection. These are called `DBClientData` modules. For more information, see [What Are DBClientData Modules in Administering JDBC Data Sources for Oracle WebLogic Server](#). In this EDG, `DBClientData` type of module is used to maintain the database client information. However, wallets and SSL configuration is not used to access the database so the `DBClientData` module contains only the appropriate `tnsnames.ora`.

The following steps are required to use a TNS alias in the different Datasources used by FMW and WLS schemas:

1. Create a `tnsnames.ora` with the pertaining alias and mapping URLs used in the connection pools. Copy the connect string from one of the existing datasource configuration files. For example,

 **Note:**

This is an example using the short jdbc URL.

```
[oracle@soahost1~]$ grep url /u01/oracle/config/domains/soaedgdomain/
config/jdbc/opss-datasource-jdbc.xml
  <url>jdbc:oracle:thin:@drdbrac12a-
scan.dbsubnet.vcnlon80.oraclevcn.com:1521/soaedg.example.com</url>
[oracle@soahost1~]$
```

Use the information in the connect string to add a long URL entry to a `tnsnames.ora` file. Use an alias name that identifies your connection. Notice that in order to deploy the `tnsnames.ora` as `DBClient` module the location of the deployment module needs to be two levels down under the domain config directory if it resides on the WLS Administration

Server node. The file can also be created in the node that runs the WebLogic Remote Console and can also be uploaded (as an application ear or war file).

```
[oracle@soahost1~]$ cat /u01/oracle/config/tnsadmin/tnsnames.ora

soaedg_alias =
  (DESCRIPTION=
    (ADDRESS_LIST=
      (LOAD_BALANCE=ON)
      (ADDRESS=(PROTOCOL=TCP) (HOST= drdbrac12a-
scan.dbsubnet.vcnlon80.oraclevcn.com) (PORT=1521)))
    (CONNECT_DATA=(SERVICE_NAME=soaedg.example.com))
  )
```

2. Deploy the directory containing the `tnsnames.ora` as a DBClientData module.
 - a. Access the domain provider in the WebLogic Remote Console.
 - b. Click **Edit Tree**.
 - c. Click **Environment > Deployments > Database Client Data Directories**.
 - d. Click **New**.

- e. Enter a name for the dbclient directory deployment. For example, `dbclientdata_modulename`.

If the directory containing the `tnsnames.ora` file resides on your local computer, uncheck the **Upload** checkbox.

- f. Click **Create**.
 - g. Click **Save**.

The cart on the top right part of the screen will display full with a yellow bag inside.

- h. Click the **Cart** icon and select **Commit Changes**.

This will create a `tnsnames/dbclient` module under domain dir `/u01/oracle/config/domains/soaedgdomain/config/dbclientdata/dbclientdata_modulename`.

You can also perform the deployment of a database client module using the `deploy` command in `wlst`.

3. Update the different Datasources and `fmwconfig` files to use the alias instead of the explicit URLs.

 **Note:**

To update a datasource to use the tns alias, the datasource configuration needs to include both a pointer to the `tnsnames.ora` file and the alias itself in the jdbc URL.

You must perform the following steps to include a pointer to the `tnsnames.ora` file include the property `oracle.net.tns_admin` in the datasource properties.

- a. Access the domain provider in the WebLogic Remote Console.
- b. Click **Edit Tree**.

- c. Click **Services > Datasources > Datasource_name**.
- d. In the navigation tree on the left, select **Properties** for the precise Datasource.
- e. Click **New**.
- f. Enter `oracle.net.tns_admin` as the property name.
- g. Click **Create**.
- h. In the next screen with the property details, enter as **value** the directory for the `dbclientdata_modulename` that is `/u01/oracle/config/domains/soaedgdomain/config/dbclientdata/dbclientdata_modulename` in the example above.
- i. Click **Save**.

The cart on the top right part of the screen will display full with a yellow bag inside.

- j. In the navigation tree on the left, click the **Datasource** name.
- k. Select the **Connection Pool** tab.
- l. In the URL, replace the URL with the alias syntax as shown below:

```
jdbc:oracle:thin:@soaedg_alias
```

- m. Click **Save**.
- n. Click the **Cart** icon and select **Commit Changes**.

If you check the datasource configuration file, it should reflect the following under the `<jdbc-driver-params>` `<properties>` entries:

```
<property>
<name>oracle.net.tns_admin</name>
<value>/u01/oracle/config/domains/soaedgdomain/config/dbclientdata/
dbclientdata_modulename</value>
</property>
```

The datasource configuration file should reflect as JDBC URL under `<jdbc-driver-params>` as shown below:

```
<url>jdbc:oracle:thin:@soaedg_alias</url>
```

4. To update the FMW jps config to use the tns alias, the `domain_path/config/fmwconfig/jps-config.xml` and `domain_path/config/fmwconfig/jps-config-jse.xml` files need to be updated and both a pointer to the `tnsnames.ora` file and the alias itself must be included in the jdbc url, that is replace the information in the `propertySet` for the DB with the updated URL and the `tnsadmin` pointer.

```
<property name="oracle.net.tns_admin" value="/u01/oracle/config/domains/
soaedgdomain/config/dbclientdata/dbclientdata_modulename "/>
<property name="jdbc.url" value="jdbc:oracle:thin:@soaedg_alias "/>
```

Restart the Administration Server for all the changes to be applied.

Alternatively, you can use the https://github.com/oracle-samples/maa/tree/main/1412EDG/fmw1412_change_to_tns_alias.sh script instead of the steps 1, 2, 3 and 4 to deploy the corresponding `DBClientData` module and replace all urls in the jdbc and jps configuration with the pertaining alias.

However, using the script is only recommended when all domain extensions have been completed and all the required datasources are present in the domain configuration because the script is configured to exit if an existing tnsadmin already exists in the configuration files. This behavior is intentional to avoid conflicts with other DBClient modules in the domain.

The recommended approach is to configure your domain to suit your functional needs (SOA, OSB, SOA+OSB, SOA+BPMN and so on as described in the [Flow Charts and Road Maps for Implementing the Primary Oracle SOA Suite Enterprise Topologies](#) section in the *About the Oracle SOA Suite Enterprise Deployment Topology* chapter). After your domain is complete and working, use the script to make the TNS alias change. Ensure you read the script's instructions in its header for its correct execution.

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 SOA Suite enterprise deployment.

 **Note:**

Some of the static and runtime artifacts listed in this section are hosted from Network Attached Storage (NAS). If possible, backup and recover these volumes from the NAS filer directly rather than from the application servers.

For general information about backing up and recovering Oracle Fusion Middleware products, see the following sections in *Administering Oracle Fusion Middleware*:

- Backing Up Your Environment
- Recovering Your Environment

[Table 19-4](#) lists the static artifacts to back up in a typical Oracle SOA Suite enterprise deployment.

Table 19-4 Static Artifacts to Back Up in the Oracle SOA Suite 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	SOAHOST1 and SOAHOST2 (or NAS Filer)	Application Tier
Installation-related files	WEBHOST1, WEHOST2, and shared storage	N/A

[Table 19-5](#) lists the runtime artifacts to back up in a typical Oracle SOA Suite enterprise deployment.

Table 19-5 Run-Time Artifacts to Back Up in the Oracle SOA Suite Enterprise Deployment

Type	Host	Tier
Administration Server domain home (ASERVER_HOME)	SOAHOST1 (or NAS Filer)	Application Tier
Application home (APPLICATION_HOME)	SOAHOST1 (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)	SOAHOST1 (or NAS Filer)	Application Tier
OHS/OTD Configuration directory	WEBHOST1 and WEBHOST2	Web Tier

- [Online Domain Run-Time Artifacts Backup/Recovery Example](#)

Online Domain Run-Time Artifacts Backup/Recovery Example

This section describes an example procedure to implement a backup of the WebLogic domain artifacts. This approach can be used during the EDG configuration process, for example, before extending the domain to add a new component.

This example has the following features:

- App tier Runtime Artifacts are backed up/recovered in this example:

Artifact	Host	Tier
Administration Server domain home (ASERVER_HOME)	SOAHOST1 (or NAS Filer)	Application Tier
Application home (APPLICATION_HOME)	SOAHOST1 (or NAS Filer)	Application Tier
Deployment Plan home (DEPLOY_PLAN_HOME)	SOAHOST1 (or NAS Filer)	Application Tier
Runtime artifacts (adapter control files) (ORACLE_RUNTIME)	SOAHOST1 (or NAS Filer)	Application Tier
Scripts and Customizations	Per host	Application Tier

- This backup procedure is suitable for cases when a major configuration change is done to the domain (that is, domain extension). If something goes wrong, or if you make incorrect selections, you can restore the domain configuration to the earlier state. Database backup/restore is not mandatory for this sample procedure, but steps to backup/restore the database are included as optional.

Artifact	Host	Tier
Oracle RAC database (optional)	Oracle RAC database (optional)	Data Tier

- Operating system tools are used in this example. Some of the run-time artifacts listed in this section are hosted from Network Attached Storage (NAS). If possible, do the backup and recovery of these volumes from the NAS filer directly rather than from the application servers.

- Managed servers are running during the backup. MSERVER_HOME is not backed up and pack/unpack procedure is used later to recover MSERVER_HOME. Therefore, managed server lock files are not included in the backup.
- AdminServer can be running during the backup if .lok files are excluded from the backup. To avoid an inconsistent backup, do not make any configuration changes until the backup is complete. To ensure that no changes are made in the WebLogic Server domain, you can lock the WebLogic Server configuration.

 **Note:**

Excluding these:

- AdminServer/data/ldap/ldapfiles/EmbeddedLDAP.lok
- AdminServer/tmp/AdminServer.lok

- [Back Up the Domain Run-Time Artifacts](#)
- [Restore the Domain Run-Time Artifacts](#)

Back Up the Domain Run-Time Artifacts

To backup the domain runtime artifacts, perform the following steps:

1. Log in to SOAHOST1 with user *oracle* and ensure that you define and export the following variables:

Variable	Example Value	Description
<i>BAK_TAG</i>	BEFORE_BPM	Descriptive tag used in the names of the backup files and database restore point.
<i>BAK_DIR</i>	/backups	Host folder where backup files are stored.
<i>DOMAIN_NAME</i>	soaedg_domain	Domain name

For example:

```
export BAK_TAG=BEFORE_BPM
export DOMAIN_NAME=soaedg_domain
export BAK_DIR=/backups
```

2. Ensure that the following domain variables are set with the values of the domain:

Variable	Example Value
<i>ASERVER_HOME</i>	/u01/oracle/config/domains/ soaedg_domain
<i>DEPLOY_PLAN_HOME</i>	/u01/oracle/config/dp
<i>APPLICATION_HOME</i>	/u01/oracle/config/applications/ soaedg_domain
<i>ORACLE_RUNTIME</i>	/u01/oracle/runtime

See [Table 7-2](#).

3. Before you make the backup, lock the domain configuration, so you prevent other accounts from making changes during your edit session. To lock the domain configuration from Fusion Middleware Control:
 - a. Log in to `https://admin.example.com:445/em`.
 - b. Locate the Change Center at the top of Fusion Middleware Control.
 - c. From the **Changes** menu, select **Lock & Edit** to lock the configuration edit for the domain.



Note:

To avoid an inconsistent backup, do not make any configuration changes until the backup is complete.

4. Log in to SOAHOST1 and clean the logs and backups applications before the backup:

```
find ${ASERVER_HOME}/servers/AdminServer/logs -type f -name "*.out0*" ! -
size 0c -print -exec rm -f {} \+
find ${ASERVER_HOME}/servers/AdminServer/logs -type f -name "*.log0*" ! -
size 0c -print -exec rm -f {} \+
find ${APPLICATION_HOME} -type f -name "*.bak*" -print -exec rm -f {} \;
```

5. Perform the backup of each artifact by using tar:

```
tar -cvzf ${BAK_DIR}/backup_aserver_home_${DOMAIN_NAME}_${BAK_TAG}.tgz $
{ASERVER_HOME} --exclude ".lok"

tar -cvzf ${BAK_DIR}/backup_dp_home_${DOMAIN_NAME}_${BAK_TAG}.tgz $
{DEPLOY_PLAN_HOME}/${DOMAIN_NAME}

tar -cvzf ${BAK_DIR}/backup_app_home_${DOMAIN_NAME}_${BAK_TAG}.tgz $
{APPLICATION_HOME}

tar -cvzf ${BAK_DIR}/backup_runtime_${DOMAIN_NAME}_${BAK_TAG}.tgz $
{ORACLE_RUNTIME}/${DOMAIN_NAME}

ls --format=single-column ${BAK_DIR}/backup_aserver_*.tgz
ls --format=single-column ${BAK_DIR}/backup_dp_*.tgz
ls --format=single-column ${BAK_DIR}/backup_app_*.tgz
ls --format=single-column ${BAK_DIR}/backup_runtime_*.tgz
```

6. Release the domain lock.
 - a. Log in to `https://admin.example.com:445/em`.
 - b. Locate the Change Center at the top of Fusion Middleware Control.
 - c. From the **Changes** menu, select **Release Configuration** to release the configuration edit for the domain.
7. Backup your scripts and customizations, if needed.
8. **(Optional)** Log in to the database and create a flashback database restore point:

 **Note:**

Flash database technology is used in this example for database recovery. Check your database version's documentation for more information about Flashback.

a. Create flashback guaranteed checkpoint.

```
sqlplus / as sysdba
SQL> create restore point BEFORE_BPM guarantee flashback database;
SQL> alter system switch logfile;
```

b. Verify.

```
SQL> set linesize 300
SQL> column name format a30
SQL> column time format a32
SQL> column storage_size format 999999999999
SQL> SELECT name, guarantee_flashback_database, time, storage_size FROM
v$restore_point ORDER BY time;
```

Example:

NAME	GUA	TIME
STORAGE_SIZE		
SOAEDG_BEFORE_BPM	YES	12-MAY-17 03.29.28.000000000 AM
8589934592		

exit

Restore the Domain Run-Time Artifacts

To recover the domain to the point where the backups were made, follow these steps:

1. Log in to SOAHOST1 using the oracle user.
2. Stop all the servers in the domain, including the AdminServer.

```
${ORACLE_COMMON_HOME}/common/bin/wlst.sh
connect('weblogic_admin_username','password','t3://adminvhn:7001')
```

```
shutdown('OSB_Cluster', 'Cluster', force='true')
shutdown('ESS_Cluster', 'Cluster', force='true')
shutdown('BAM_Cluster', 'Cluster', force='true')
shutdown('MFT_Cluster', 'Cluster', force='true')
shutdown('SOA_Cluster', 'Cluster', force='true')
shutdown('WSM-PM_Cluster', 'Cluster', force='true')
```

```
state('SOA_Cluster', 'Cluster')
state('OSB_Cluster', 'Cluster')
state('ESS_Cluster', 'Cluster')
state('BAM_Cluster', 'Cluster')
state('MFT_Cluster', 'Cluster')
state('WSM-PM_Cluster', 'Cluster')
```



```
shutdown('AdminServer', force='true', block='true')
```

3. Ensure that the following domain variables are set with the values of the domain:

Variable	Example Value
<i>ASERVER_HOME</i>	/u01/oracle/config/domains/soaedg_domain
<i>DEPLOY_PLAN_HOME</i>	/u01/oracle/config/dp
<i>APPLICATION_HOME</i>	/u01/oracle/config/applications/soaedg_domain
<i>ORACLE_RUNTIME</i>	/u01/oracle/runtime

4. Remove the current folders by renaming them. You can remove these folders completely at the end of the process after you have verified the recovered domain.

- a. In SOAHOST1:

```
mv ${ASERVER_HOME} ${ASERVER_HOME}_DELETE
mv ${DEPLOY_PLAN_HOME}/${DOMAIN_NAME} ${DEPLOY_PLAN_HOME}/${DOMAIN_NAME}_DELETE
mv ${APPLICATION_HOME} ${APPLICATION_HOME}_DELETE
mv ${ORACLE_RUNTIME}/${DOMAIN_NAME} ${ORACLE_RUNTIME}/${DOMAIN_NAME}_DELETE
```

- b. In each SOAHOSTN:

```
mv ${MSERVER_HOME} ${MSERVER_HOME}_DELETE
```

5. Locate and identify the backups in the backup folder. Ensure that you define and export the following variables with the correct values of the backup you want to recover:

Variable	Example Value	Description
<i>BAK_TAG</i>	BEFORE_BPM	Descriptive tag used in the names of the backup files and database restore point.
<i>BAK_DIR</i>	/backups	Host folder where backup files are stored.
<i>DOMAIN_NAME</i>	soaedg_domain	Domain name

For example:

```
export BAK_TAG=BEFORE_BPM
export DOMAIN_NAME=soaedg_domain
export BAK_DIR=/backups
```

6. Perform the recovery of the files by extracting the files.

 **Note:**

TAR files will recreate the structure beginning with /, so you need to go to / folder.

```
cd /
tar -xzf ${BAK_DIR}/backup_aserver_home_${DOMAIN_NAME}_${BAK_TAG}.tgz
tar -xzf ${BAK_DIR}/backup_dp_home_${DOMAIN_NAME}_${BAK_TAG}.tgz
tar -xzf ${BAK_DIR}/backup_app_home_${DOMAIN_NAME}_${BAK_TAG}.tgz
tar -xzf ${BAK_DIR}/backup_runtime_${DOMAIN_NAME}_${BAK_TAG}.tgz
```

7. (Optional) If you need to recover the database to the flashback recovery point, perform the following steps:

- a.** Log in to DBHOST with oracle user and stop the database:

```
srvctl stop database -database soaedgdb
```

- b.** Log in to the database and flashback database to the restore point:

```
sqlplus / as sysdbaSQL>
startup mountSQL>
FLASHBACK DATABASE TO RESTORE POINT BEFORE_BPM;
Flashback complete.
```

- c.** Start database with this command:

```
SQL> ALTER DATABASE OPEN RESETLOGS;
```

8. Start AdminServer:

```
${ORACLE_COMMON_HOME}/common/bin/wlst.sh
wls:/offline> nmConnect('nodemanager','password','ADMINVHN','5556',
'domain_name','ASERVER_HOME','PLAIN')
Connecting to Node Manager ...
Successfully Connected to Node Manager.
wls:/nm/domain_name > nmStart('AdminServer')
```

9. Propagate the domain to the Managed Servers.

- a.** Sign in to SOAHOST1 and run the pack command to create the template, as follows:

```
cd ${ORACLE_COMMON_HOME}/common/bin
./pack.sh -managed=true
-domain=ASERVER_HOME \
-template=/full_path/recover_domain.jar \
-template_name=recover_domain_template \
-log_priority=DEBUG \
-log=/tmp/pack.log
```

- 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 where you want to create the domain template jar file.
- `recover_domain.jar` is an example of the name for the jar file that you are creating.
- `recover_domain_template` is an example of the name for the jar file that you are creating.

b. Run the `unpack` command in every SOAHOST, as follows:

```
cd $ORACLE_COMMON_HOME/common/bin

./unpack.sh -domain=MSERVER_HOME \
            -overwrite_domain=true \
            -template=/full_path/recover_domain.jar
            -log_priority=DEBUG \
            -log=/tmp/unpack.log \
            -app_dir=APPLICATION_HOME \
```

- 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 `/full_path/ recover_domain.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device.

10. Recover/perform customizations, if needed.

11. Start the servers and verify the domain.

12. After checking that everything is correct, you can delete the previous renamed folders:

a. In SHOAHOST1:

```
rm -rf    ${ASERVER_HOME}_DELETE
rm -rf    ${KEYSTORE_HOME}_DELETE
rm -rf    ${DEPLOY_PLAN_HOME}/${DOMAIN_NAME}_DELETE
rm -rf    ${APPLICATION_HOME}_DELETE
rm -rf    ${ORACLE_RUNTIME}/${DOMAIN_NAME}_DELETE
```

b. In every SOAHOSTN:

```
rm -rf    ${MSERVER_HOME}_DELETE
```

Configuration and Management Tasks for an Oracle SOA Suite Enterprise Deployment

These are some of the key configuration and management tasks that you likely need to perform on an Oracle SOA Suite enterprise deployment.

- [Deploying Oracle SOA Suite Composite Applications to an Enterprise Deployment](#)
- [Using Shared Storage for Deployment Plans and SOA Infrastructure Applications Updates](#)
- [Managing Database Growth in an Oracle SOA Suite Enterprise Deployment](#)

- [Managing the JMS Messages in a SOA Server](#)

Deploying Oracle SOA Suite Composite Applications to an Enterprise Deployment

Oracle SOA Suite applications are deployed as composites, consisting of different kinds of Oracle SOA Suite components. SOA composite applications include the following:

- Service components such as Oracle Mediator for routing, BPEL processes for orchestration, BAM processes for orchestration (if Oracle BAM Suite is also installed), human tasks for workflow approvals, spring for integrating Java interfaces into SOA composite applications, and decision services for working with business rules.
- Binding components (services and references) for connecting SOA composite applications to external services, applications, and technologies.

These components are assembled into a single SOA composite application.

When you deploy an Oracle SOA Suite composite application to an Oracle SOA Suite enterprise deployment, be sure to deploy each composite to a specific server or cluster address and not to the load balancer address (`soa.example.com`).

Deploying composites to the load balancer address often requires direct connection from the deployer nodes to the external load balancer address. As a result, you have to open additional ports in the firewalls.

For more information about Oracle SOA Suite composite applications, see the following sections in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*:

- [Deploying SOA Composite Applications](#)
- [Monitoring SOA Composite Applications](#)
- [Managing SOA Composite Applications](#)

Using Shared Storage for Deployment Plans and SOA Infrastructure Applications Updates

When you redeploy a SOA infrastructure application or resource adapter within the SOA cluster, the deployment plan along with the application bits should be accessible to all servers in the cluster.

SOA applications and resource adapters are installed using nostage deployment mode. Because the administration sever does not copy the archive files from their source location when the nostage deployment mode is selected, each server must be able to access the same deployment plan.

To ensure deployment plan location is available to all servers in the domain, use the Deployment Plan home location described in [File System and Directory Variables Used in This Guide](#) and represented by the `DEPLOY_PLAN_HOME` variable in the *Enterprise Deployment Workbook*.

Managing Database Growth in an Oracle SOA Suite Enterprise Deployment

When the amount of data in the Oracle SOA Suite database grows very large, maintaining the database can become difficult, especially in an Oracle SOA Suite enterprise deployment where potentially many composite applications are deployed.

See the following sections in *Administering Oracle SOA Suite and Oracle Business Process Management Suite*:

- Developing a Database Growth Management Strategy
- Managing Database Growth

Managing the JMS Messages in a SOA Server

There are several procedures to manage JMS messages in a SOA server. You may need to perform these procedures in some scenarios, for example, to preserve the messages during a scale-in operation.

This section explains some of these procedures in detail.

- [Draining the JMS Messages from a SOA Server](#)
- [Importing the JMS Messages into a SOA Server](#)

Draining the JMS Messages from a SOA Server

The process of draining the JMS messages helps you clear out the messages from a particular WebLogic server. A basic approach to drain stores consists of stopping the message production in the appropriate JMS Servers and allowing the applications to consume the messages.

This procedure, however, is application dependent, and could take an unpredictable amount of time. As an alternative, general instructions are provided here for saving the current messages from their current JMS destinations and, when/if required, importing them into a different server.

The draining procedure is useful in scale-in/down scenarios, where the size of the cluster is reduced by removing one or more servers. You can ensure that no messages are lost by draining the messages from the server that you delete, and then importing them into another server in the cluster.

You can also use this procedure in some disaster recovery maintenance scenarios, when the servers are started in a secondary location by using an Snapshot Standby database. In this case, you may need to drain the messages from the domain before starting it in the secondary location to avoid their consumption in the standby domain when you start the domain (otherwise, duplicate executions could take place). You cannot import messages in this scenario.

To drain the JMS messages from a server, perform the following steps:

1. Stop a new workload by pausing production for the JMS Server. You must do this activity for each JMS Server of the server that is affected in the operation:
 - a. In the WebLogic Remote Console, open the **Monitoring Tree**.
 - b. Navigate to **Environment > Servers**.
 - c. In the server that you want to delete, navigate to the **Services > Messaging > JMS Runtime > JMS Servers**.
 - d. Select the **JMS Servers**.
 - e. Click **Pause > Production**.
2. Drain the messages from the destinations. To drain the JMS messages, you can let applications consume the pending messages. However, this task is application dependent

and may take time. Hence, Oracle recommends you to export the messages of each destination. Verify which destinations have messages:

- a. In the WebLogic Remote Console, open the **Monitoring Tree**.
 - b. Navigate to **Environment > Servers**.
 - c. In the server that you want to delete, navigate to **Services > Messaging > JMS Runtime > JMS Servers**.
 - d. For each JMS Server, look whether the destination members have current messages. Identify the destination name, its **JMS Module** and **JMS Server**.
 - e. Repeat this activity for each JMS Server that is running in the server that you want to delete.
- **Drain messages from queues:** For those queue destinations that have current messages:
 - a. In the WebLogic Remote Console, open the **Monitoring Tree**.
 - b. Navigate to **Dashboards** and click **JMS Destinations** dashboard.
 - c. Select the queue that you want to export messages from.
 - d. In the **Messages** tab, select **Export > Export All** and export the messages to a file. Make a note of the file name for later use.
 - e. Delete the exported messages by using the **Delete All** option. This step is important to avoid message duplications.
 - **Drain messages from topics**

Oracle recommends you to drain and import messages from topics only if they have a critical business impact. See [Table 19-6](#) for details about the purpose and business impact for each topic. Only the loss of messages in the topic **dist_EDNTopic_auto**, used by EDN, has a business impact.

Table 19-6 Details of the Purpose and Business Impact for Each Topic of a Component

Component	JMS Module	JMS Topic Name	Purpose	Business Impact of Message Loss
BPM	BPMJMSModule	dist_MeasurementTopic_auto	Used for publishing process metrics messages to the internal process star schema.	Low impact. Will affect some dashboard number appearing in the PCS workspace dashboards and BAM dashboards based on the process star schema data object.
BPM	BPMJMSModule	dist_PeopleQueryTopic_auto	Used for updating logical group memberships.	Low impact. The group membership will be recalculated based on a scheduler.

Table 19-6 (Cont.) Details of the Purpose and Business Impact for Each Topic of a Component

Component	JMS Module	JMS Topic Name	Purpose	Business Impact of Message Loss
SOA	SOAJMSModule	dist_B2BBroadcastTopic_auto	Used by B2B, messages are meant to be consumed immediately.	No impact.
SOA	SOAJMSModule	dist_EDNTopic_auto	Used for EDN, contains event messages for applications.	Business impact. Applications that consume these EDN event messages will lose them.
SOA	SOAJMSModule	dist_TenantTopic_auto	No longer used.	No impact.
SOA	SOAJMSModule	dist_XmlSchemaChangeNotificationTopic_auto	No longer used.	No impact.
Insight	ProcMonJMSModule	dist_ProcMonActivationTopic_auto	Used by Insight for lifecycle operations - for activating an insight model across different nodes of the cluster.	No impact.
BAM	BAMJMSSystem Resource	dist_oracle.beam.cqs.active_data_auto	Not used in production.	No impact.
BAM	BAMJMSSystem Resource	dist_oracle.beam.persistence.active_data_auto	Data change notifications sent from persistence to the continuous query processor in support of active-data queries.	Low impact. Message loss could only cause incorrect data to be displayed in the active-data dashboards. Refreshing the dashboards or restarting the active-query will restore the correct data.
BAM	BAMJMSSystem Resource	dist_oracle.beam.server.event.reportcache.changelist_auto	Data changes sent from the report cache to the active-data dashboards.	Low impact. Message loss could only cause incorrect data to be displayed in the active-data dashboards. Refreshing the dashboards or restarting the active-query will restore the correct data.
BAM	BAMJMSSystem Resource	dist_oracle.beam.server.metadatachange_auto	Metadata changes sent to the downstream listeners if artifacts (queries, views, dashboards) are modified.	Low impact. Message loss could only cause incorrect data to be displayed in the active-data dashboards. Refreshing the dashboards or restarting the active-query will restore the correct data.

Table 19-6 (Cont.) Details of the Purpose and Business Impact for Each Topic of a Component

Component	JMS Module	JMS Topic Name	Purpose	Business Impact of Message Loss
MFT	MFTJMSModule	dist_MFTSystemEventTopic_auto	Used for publishing events that require synchrony in all the nodes, such as activation of the listening source, adding the PGP key, Mbean property changes, and so on.	Low impact. These messages are very short lived and their frequency is low. If there is any message loss, a restart ensures that all nodes in sync.

Follow these steps drain messages from the topics:

- a. In the WebLogic Remote Console, open the **Monitoring Tree**.
- b. Navigate to **Dashboards** and click the **JMS Destinations** dashboard.
- c. Select the topic that you want to delete and navigate to its **Subscribers**.
- d. Select the Durable Subscriber that has current messages and click **Show Messages**.
- e. Click **Export > Export All** and export the messages to a file. Make a note of the file name for later use.
- f. Delete the exported messages from the subscriber by clicking **Delete > Delete All**. This step is important to avoid message duplications.
- g. Repeat the export process for any subscriber in the topic that has current messages.

Importing the JMS Messages into a SOA Server

Messages that have been previously exported can be imported in another or the same member of the JMS destination. This procedure is used in scale-in/down scenarios, to import the messages from the server that you want to remove, to another member in the cluster.

To import the JMS messages, perform the following steps:

- **Import messages in a queue:**
 1. In the WebLogic Remote Console, open the **Monitoring Tree**.
 2. Navigate to **Dashboards** and click the **JMS Destinations** dashboard.
 3. Select the queue where you want to import messages.
 4. In the **Messages** tab, select **Import** to import the messages of this destination.
 5. Repeat the steps for each queue destination.
- **Import messages in a topic:**
 1. In the WebLogic Remote Console, open the **Monitoring Tree**.
 2. Navigate to **Dashboards** and click the **JMS Destinations** dashboard.
 3. Choose the topic member where you want to import the messages.

4. In the topic, expand the durable subscribers and select the one where you want to import the messages.
5. Click **Show Messages** and **Import**. Select the file with the messages of this subscriber.
6. Repeat the steps for each subscriber in the topic where you have to import messages.

Using Service Migration in an Enterprise Deployment

The Oracle WebLogic Server migration framework supports Whole Server Migration and Service Migration. Whole Server Migration requires more resources and a full start of a managed server, so it involves a higher failover latency than Service Migration. The products included in this EDG support Service Migration. Hence, Service Migration is recommended and this guide explains how to use Service Migration in an Oracle Fusion Middleware enterprise topology. Whole Server migration is out of the scope of this guide.

- [About Automatic Service Migration in an Enterprise Deployment](#)
Oracle WebLogic Server provides a migration framework that is an integral part of any highly available environment. The following sections provide more information about how this framework can be used effectively in an enterprise deployment.
- [Creating a GridLink Data Source for Leasing](#)
Automatic Service Migration require a data source for the leasing table, which is a table that resides in a tablespace and schema created automatically as part of the Oracle WebLogic Server schemas by the Repository Creation Utility (RCU).
- [Configuring Automatic Service Migration in an Enterprise Deployment](#)
The services used by the different SOA components in this Enterprise Deployment Guide are already configured with Automatic Service Migration when following the Configuration Wizard steps provided in this guide. For any other custom services, you can use the following steps to configure service migration.

About 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 Service Migration in an Enterprise Deployment](#)
- [Understanding Which Products and Components Require 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 virtual hostname mapping to 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*.

Whole Server Migration requires more resources (virtual IP and virtual hostname) than Automatic Service Migration. It also involves a full start of the migrated managed server, so it provides a worse recovery time objective. All the different components in Oracle FMW SOA Suite support Automatic Service Migration and this is the recommend failover approach recommended and described in this Enterprise Deployment Guide. Whole Server migration is out of the scope of this guide.

Implications of Service Migration in an Enterprise Deployment

Using Automatic Service Migration (ASM) in an Enterprise Deployment has implications in the infrastructure and configuration requirements.

The implications are:

- The resources used by servers must be accessible to both the original and failover system
In its initial status, resources are accessed by the original server or service. When a server or service is failed over/restarted in another system, the same resources (such as external resources, databases, and stores) must be available in the failover system. Otherwise, the service cannot resume the same operations. It is for this reason, that both whole server and service migration require that all members of a WebLogic cluster have access to the same transaction and JMS persistent stores.
Oracle allows you to use JDBC stores, which leverage the consistency, data protection, and high availability features of an oracle database and makes resources available for all the servers in the cluster. When you configure persistent stores properly in the database you must ensure that if a failover occurs (whole server migration or service migration), the failover system is able to access the same stores without any manual intervention.
- Leasing Datasource
Service migration requires the configuration of a leasing datasource that is used by servers to store *alive* timestamps. These timestamps are used to determine the health of a server or service, and are key to the correct behavior of server and service migration (they are used to marks servers or services as *failed* and trigger failover).

 **Note:**

Oracle does not recommend that you use consensus leasing for HA purposes.

[Summary of Aspects of ASM](#) summarizes the different aspects.

Table 20-1 Summary of Aspects of ASM

Cluster Protection	Failover Time	Capacity Planning	Reliability	Shared Storage/DB	VIP per Managed Server
ASM	30 secs	Mem/CPU of services	DB Leasing	Yes	No

Understanding Which Products and Components Require Service Migration

The following table lists the recommended best practice for optimal High Availability using Automatic Service Migration. As explained in the previous sections, Automatic Server Migration can also be used with all these components but it provides a worse failover time and requires more resources.

Component	Automatic Service Migration (ASM)
Oracle Web Services Manager (OWSM)	NO
Oracle SOA Suite	YES
Oracle Service Bus	YES
Oracle Business Process Management	YES
Oracle Enterprise Scheduler	NO
Oracle Business Activity Monitoring	YES
Oracle B2B	YES
Managed File Transfer	YES

Creating a GridLink Data Source for Leasing

Automatic Service Migration require a data source for the leasing table, which is a table that resides in a tablespace and schema 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 `WLSRuntimeSchemaDataSource` as is for database leasing. This datasource is already configured with the `FMW1412_WLS_RUNTIME` schema, where the leasing table is stored.

For an enterprise deployment, you should create a GridLink data source:

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Services** and select **Data Sources**.
4. On the Summary of **Data Sources** page, click **New** and select **GridLink Data Source**. Enter the following:

Table 20-2 GridLink Data Source Properties

Properties	Description
Name	Enter a logical name for the data source in the Name field. For example, Leasing.
JNDI Names	Enter a name for JNDI. For example, jdbc/leasing.
Targets	Select the cluster that you are configuring for Automatic Service Migration and move to "Chosen".
Data Source Type	Select GridLink Data Source .
Database Driver	Select Oracle's Driver (Thin) for GridLink Connections Versions: Any .
Global Transaction Protocol	Select None .
Listeners	Enter the SCAN address and port for the RAC database, separated by a colon. For example, db-scan.example.com:1521.
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, soaedg.example.com.
Database username	Enter the user name of the WLS Runtime schema. For example, FMW1412_WLS_RUNTIME. In this example, FMW1412 is the prefix you used when you created the schemas as you prepared to configure the domain.
Password	Enter the password you used when you created the WLS schema in RCU.
Protocol	Leave the default value (TCP).
Fan Enabled	You must check this option.
ONS Nodes	Leave it empty. ONS node list is automatically retrieved when the database is 12.2 or higher version.
ONS Wallet and password	You can leave this field empty.
Test Configuration	You must enable this option.

 **Note:**

The leasing table is created automatically when you create the WLS schemas with the Repository Creation Utility (RCU).

5. Click **Create**.

6. Commit changes in the shopping cart.

Configuring Automatic Service Migration in an Enterprise Deployment

The services used by the different SOA components in this Enterprise Deployment Guide are already configured with Automatic Service Migration when following the Configuration Wizard steps provided in this guide. For any other custom services, you can use the following steps to configure service migration.

- [Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster](#)
- [Configuring Automatic Service Migration](#)

Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster

Before you can configure automatic service migration, you must verify the leasing mechanism and data source that is used by the automatic service migration feature.

Note:

To accomplish data source consolidation and connection usage reduction, you can reuse the `WLSRuntimeSchemaDataSource` datasource as is for database leasing. This datasource is already configured with the `FMW1412_WLS_RUNTIME` schema, where the leasing table is stored.

The following procedure assumes that you have configured the Leasing data source either by reusing the `WLSRuntimeSchemaDataSource` or a custom datasource that you created as described in [Creating a GridLink Data Source for Leasing](#).

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Environment > Clusters**.
4. The Summary of **Clusters** page appears. Click the cluster for which you want to configure migration.
5. Click the **Migration** tab.
6. Verify that Database is selected in the **Migration Basis** drop-down menu.
7. In the **Data Source for Automatic Migration** drop-down menu, select the Leasing data source that you created in *Creating a GridLink Data Source for Leasing*. Select the `WLSRuntimeSchemaDataSource` for data source consolidation.
8. Click **Save**.
9. Commit changes in the shopping cart
10. Restart the managed servers for the changes to be effective. If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

After you complete the database leasing configuration, continue with the configuration of the service migration:

- See [Configuring Automatic Service Migration](#)

Configuring Automatic Service Migration

After you have configured the leasing for the cluster as described in [Setting the Leasing Mechanism and Data Source for an Enterprise Deployment Cluster](#), you can configure automatic service migration for specific services in an enterprise deployment. The following sections explain how to configure and validate Automatic Service Migration for static clusters.

- [Changing the JTA Migration Settings for the Managed Servers in the Cluster](#)
- [About Selecting a Service Migration Policy](#)
- [Setting the Service Migration Policy for Each Managed Server in the Cluster](#)
- [Validating Automatic Service Migration](#)
- [Failing Back Services After Automatic Service Migration](#)

Changing the JTA Migration Settings for the Managed Servers in the Cluster

After you set the leasing mechanism and data source for the cluster, you can enable automatic JTA migration for the Managed Servers that you want to configure for service migration. Note that this topic applies only if you are deploying JTA services as part of your enterprise deployment.

To change the migration settings for the Managed Servers in each cluster:

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand the **Environment** node and click **Servers**.
The **Summary of Servers** page appears.
4. Expand the name of the server you want to modify.
5. Navigate to **JTA Migratable Target**.
6. In the the **JTA Migration Policy** drop-down menu, select **Failure Recovery**.

In the **JTA Candidate Servers** section of the page, leave the **Chosen** list box empty. If you do not select any servers from the **Available** list box, all the available servers in the cluster become candidates for service migration.

7. Click **Save**.
8. Commit the changes in the shopping cart.
9. Restart the Managed Servers and the Administration Server for the changes to be effective.

If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

About Selecting a Service Migration Policy

When you configure Automatic Service Migration, you select a Service Migration Policy for each cluster. This topic provides guidelines and considerations when selecting the Service Migration Policy.

For example, products or components running singletons or using Path services can benefit from the **exactly-once** policy. With this policy, if at least one Managed Server in the candidate server list is running, the services hosted by this migratable target are active somewhere in the cluster if servers fail or are administratively shut down (either gracefully or forcibly). This can cause multiple homogenous services to end up in one server on startup.

When you use this policy, you should monitor the cluster startup to identify what servers are running on each server. You can then perform a manual failback, if necessary, to place the system in a balanced configuration.

Other Fusion Middleware components are better suited for the **failure-recovery** policy.

Based on these guidelines, the following policies are recommended for an Oracle SOA Suite enterprise topology:

- SOA_Cluster: **failure-recovery**
- OSB_Cluster: **failure-recovery**
- BAM_Cluster: **exactly-once**
- MFT_Cluster: **failure-recovery**

See Policies for Manual and Automatic Service Migration in *Administering Clusters for Oracle WebLogic Server*.

Setting the Service Migration Policy for Each Managed Server in the Cluster

After you modify the JTA migration settings for each server in the cluster, you can then identify the services and set the migration policy for each Managed Server in the cluster, using the WebLogic Remote Console:

1. Log into the WebLogic Remote Console.
2. Navigate to the **Edit Tree**.
3. In the structure tree, expand **Environment > Migratable Targets**.
4. Click the name of the first Managed Server in the cluster.
5. In the **Service Migration Policy** drop-down menu, select the appropriate policy for the cluster. See [About Selecting a Service Migration Policy](#).
6. In the **Candidate** tab, leave the **Chosen** list box empty. If you do not select any servers from the **Available** list box, all the available servers in the cluster become candidates for service migration.
7. Click **Save**.
8. Repeat *Step 2* to *Step 6* for each of the additional Managed Servers in the cluster.
9. Commit changes in the shopping cart.
10. Restart the managed servers for the changes to be effective.

If you are configuring other aspects of ASM in the same configuration change session, you can use a final unique restart to reduce downtime.

Validating Automatic Service Migration

After you configure automatic service migration for your cluster and Managed Servers, validate the configuration, as follows:

1. If you have not already done so, log into the WebLogic Remote Console.

2. Navigate to the **Monitoring Tree**.
3. In the structure tree, expand **Environment > Migration**.
4. Click **Service Migration Data Runtimes**.

The console displays a list of migratable targets and their current hosting server.

5. In the **Migratable Targets** table, select a row for one of the migratable targets.
6. Note the value in the **Migrated To** property.
7. Use the operating system command line to stop the first Managed Server.

Use the following command to end the Managed Server Process and simulate a crash scenario:

```
kill -9 pid
```

In this example, replace *pid* with the process ID (PID) of the Managed Server. You can identify the PID by running the following UNIX command:

```
ps -ef | grep managed_server_name
```

 **Note:**

After you kill the process, the Managed Server might be configured to start automatically. In this case, you must kill the second process using the `kill -9` command again.

8. Watch the terminal window (or console) where the Node Manager is running.

You should see a message indicating that the selected Managed Server has failed. The message is similar to the following:

```
<INFO> <domain_name> <server_name>  
<The server 'server_name' with process id 4668 is no longer alive; waiting for the  
process to die.>  
<INFO> <domain_name> <server_name>  
<Server failed during startup. It may be retried according to the auto restart  
configuration.>  
<INFO> <domain_name> <server_name>  
<Server failed but will not be restarted because the maximum number of restart  
attempts has been exceeded.>
```

9. Return to the WebLogic Remote Console and refresh the table of Service Migration Data Runtimes; verify that the migratable targets are transferred to the remaining, running Managed Server in the cluster:
 - Verify that the **Migrated to** value for the process you killed is now updated to show that it has been migrated to a different host.
 - Verify that the value in the **Status of Last Migration** column for the process is *Succeeded*.
10. Open and review the log files for the Managed Servers that are now hosting the services; look for any JTA or JMS errors.

 **Note:**

For JMS tests, it is a good practice to get message counts from destinations and make sure that there are no stuck messages in any of the migratable targets:

For example, for uniform distributed destinations (UDDs):

- a. Log into the **WebLogic Remote Console** and navigate to the **Monitoring Tree**.
- b. Navigate to **Dashboards** and click **JMS Destinations**.
- c. Order by Destination Name and look for the destination.

 **Tip:**

You can copy this dashboard and create a custom one to filter by specific destination name.

- d. Review the **Messages Current Count** and **Messages Pending Count** values.

Failing Back Services After Automatic Service Migration

When Automatic Service Migration occurs, Oracle WebLogic Server does not support failing back services to their original server when a server is back online and rejoins the cluster.

As a result, after the Automatic Service Migration migrates specific JMS services to a backup server during a fail-over, it does not migrate the services back to the original server after the original server is back online. Instead, you must migrate the services back to the original server manually.

To fail back a service to its original server, use WLST migrate command. For more information, see [WLST Command Reference for Oracle WebLogic Server](#).

21

Scaling Procedures for an Enterprise Deployment

The scaling procedures for an enterprise deployment include scale out, scale in, scale up, and scale down. During a scale-out operation, you add managed servers to new nodes. You can remove these managed servers by performing a scale in operation. During a scale-up operation, you add managed servers to existing hosts. You can remove these servers by performing a scale-down operation.

This chapter describes the procedures to scale out/in and scale up/down clusters.

- [Scaling Out the Topology](#)
This section lists the prerequisites, explains the procedure to scale out the topology, describes the steps to verify the scale-out process, and finally the steps to scale down (shrink).
- [Scaling in the Topology](#)
This section describes how to scale in the topology for a cluster.
- [Scaling Up the Topology](#)
This section describes how to scale up the topology.
- [Scaling Down the Topology](#)
This section describes how to scale down the topology.

Scaling Out the Topology

This section lists the prerequisites, explains the procedure to scale out the topology, describes the steps to verify the scale-out process, and finally the steps to scale down (shrink).

- [Prerequisites for Scaling Out](#)
- [Scaling Out a Cluster](#)
- [Verifying the Scale Out](#)

Prerequisites for Scaling Out

Before you perform a scale out of the topology, you must ensure that you meet the following requirements:

- The starting point is a cluster with managed servers already running.
- The new node can access the existing home directories for WebLogic Server and SOA. Use the existing installations in shared storage. You do not need to install WebLogic Server or SOA binaries in a new location. However, you do need to run `pack` and `unpack` commands to bootstrap the domain configuration in the new node.
- It is assumed that the cluster syntax is used for all internal RMI invocations, JMS adapter, and so on.

Scaling Out a Cluster

The steps provided in this procedure use the SOA EDG topology as a reference. Initially there are two application tier hosts (SOAHOST1 and SOAHOST2), each running one managed server of each cluster. A new host SOAHOST3 is added to scale up the clusters with a third managed server. `WLS_XYZn` is the generic name given to the new managed server that you add to the cluster. Depending on the cluster that is being extended and the number of existing nodes, the actual names are `WLS_SOA3`, `WLS_OSB3`, `WLS_ESS3`, and so on.

The scale-out procedure requires downtime for the existing servers in the WLS cluster being scaled if service migration has been configured for them with a different migration policy from the default one (manual). It also implies downtime if the existing migratable targets do not use an empty Candidate Server list. Using empty candidate lists is the best practice because it means that all the servers in the cluster are candidates for migration. You can check the list of candidates for each migratable targets through the WebLogic Remote console:

1. Access the domain with the WebLogic Remote Console.
2. In the left-top corner, click **Edit Tree** in the the Remote Console Screen
3. Expand **Environment** in the navigation tree.
4. Expand **Migratable targets** in the navigation tree.
5. Click each migratable target and verify the **Constrained Candidate Servers** list under **Migration** tab.

If you have created your environment following the Enterprise Deployment Guide, these lists are empty out-of-the-box. When you add a new server to the cluster, the server is automatically considered for migration without the need to restart the existing servers.

If you had decided to constraint the migration to some specific servers of the cluster only, your Candidate Server lists will not be empty. When you add a new server to the cluster, you may need to modify them to add the new server. In this case, you will have to restart the existing nodes during the scale-out process. Changing migration policy from the manual one for the new server also prompts for a restart of existing members in the cluster. Oracle recommends that you “batch” these two changes and perform one single restart later you complete both these changes (migration policy and list of candidates).

To scale out the cluster, complete the following steps:

1. On the new node, mount the existing FMW Home, which should include the SOA installation and the domain directory. Ensure that the new node has access to this directory, similar to the rest of the nodes in the domain.
2. Locate the inventory in the shared directory (for example, `/u01/oracle/products/oraInventory`), per Oracle’s recommendation. So you do not need to attach any home, but you may want to execute the script: `/u01/oracle/products/oraInventory/createCentralInventory.sh`.

This command creates and updates the local file `/etc/oraInst.loc` in the new node to point it to the `oraInventory` location.

If there are other inventory locations in the new host, you can use them, but `/etc/oraInst.loc` file must be updated accordingly for updates in each case.

3. Update the `/etc/hosts` files to add the alias `SOAHOSTn` for the new node, as described in [Verifying IP Addresses and Host Names in DNS or Hosts File](#).

For example:

```
10.229.188.204 host1-vip.example.com host1-vip ADMINVHN
10.229.188.205 host1.example.com host1 SOAHOST1
10.229.188.206 host2.example.com host2 SOAHOST2
```

```
10.229.188.207 host3.example.com host3 WEBHOST1
10.229.188.208 host4.example.com host4 WEBHOST2
10.229.188.209 host5.example.com host5 SOAHOST3
```

4. Configure a per host node manager in the new node, as described in [Creating a Per Host Node Manager Configuration](#), but do not start it yet. It will be started later.
5. Log into the WebLogic Remote Console to create a new machine:
 - a. Go to **Environment** and select **Machines**.
 - b. Click **New** to create a new machine for the new node.
 - c. Set **Name** to *SOAHOST n* (or *MFTHOST n* or *BAMHOST n*).
 - d. Click the **Node Manager** tab.
 - e. Set **Type** to **SSL**.
 - f. Set **Listen Address** to *SOAHOST n* .
 - g. Click **Save and Commit changes** in the **Shopping Cart**.
6. Use the Oracle WebLogic Remote Console to clone the first managed server in the cluster into a new managed server.
 - a. Go to **Environment** and select **Servers**.
 - b. Click **Create** and in the **Copy settings from another server**, select the first managed server in the cluster to scale out and click **Create**.
 - c. Select the first managed server in the cluster to scale out and click **Create**.
 - d. Use [Details of the Cluster to be Scaled Out](#) to set the correspondent name, listen address, and SSL listen port depending on the cluster that you want to scale out.
 - e. Click the new managed server, select **Configuration**, and then click **General**.
 - f. Update the **Machine** from *SOAHOST1* to *SOAHOST n* .
 - g. Update the Administration port for the server also to be consistent with other server in the cluster. For example, for SOA servers use port 9004, for OSB servers use 9007, and so on. Refer to the existing servers for their appropriate Administration Port.
 - h. Click **Save and Commit changes** in the **Shopping Cart**.

Table 21-1 Details of the Cluster to be Scaled Out

Cluster to Scale Out	Server to Clone	New Server Name	Server Listen Address	SSL Server Listen Port	Local Administrative Port Override
WSM-PM_Cluster	WLS_WSM1	WLS_WSM3	SOAHOST3	7010	9003
SOA_Cluster	WLS_SOA1	WLS_SOA3	SOAHOST3	7004	9004
ESS_Cluster	WLS_ESS1	WLS_ESS3	SOAHOST3	7008	9006
OSB_Cluster	WLS_OSB1	WLS_OSB3	SOAHOST3	8003	9007
BAM_Cluster	WLS_BAM1	WLS_BAM3	SOAHOST3	7006	9005
MFT_Cluster	WLS_MFT1	WLS_MFT3	MFTHOST3	7010	9014

7. Update the deployment Staging Directory Name of the new server, as described in [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).
8. Create a new key certificate and update your domain certificate store, as described in the [Creating Certificates and Certificate Stores for the WebLogic Domain](#) in *Creating the Initial Infrastructure Domain for an Enterprise Deployment* chapter. For adding only a new address (instead of all the ones detected in config.xml) you can use the `generate_perdomainCACERTS.sh` script with the following syntax:

```
./generate_perdomainCACERTS.sh [WLS_DOMAIN_DIRECTORY] [WL_HOME]
[KEYSTORE_HOME] [KEYPASS] [NEWADDR]
```

Where `NEWADDR` is the listen address for the new server being added.

9. Your new server's keystore location and ssl configuration is carried over from the server copied (`WLS_SOA1`) but you must update the password again (since it will be encrypted again for the new server) and the "Server private key alias" entry for this new server.
 - a. Navigate to **Environment > Servers**.
 - b. Click the new server.
 - c. Navigate to **Security > Keystores**.
 - d. Update the **Custom Identity Key Store Pass Phrase** and **Custom Trust Key Store Pass Phrase** with the password provided to the `generate_perdomainCACERTS.sh` script.
 - e. Click the **SSL** tab under **Security**.
 - f. Update the **Server Private Key Pass Phrase** with the password provided to the `generate_perdomainCACERTS.sh` script
 - g. Add the listen address that you used in the previous step (certificate generation for the new server) as **Server Private Key Alias**.
10. Update the TLOG JDBC persistent store of the new managed server:
 - a. Log into the **WebLogic Remote Console**.
 - b. Go to **Environment** and expand the **Servers** link on the navigation tree on the left.
 - c. Click the new server `WLS_XYZn`.
 - d. Click the **Services > JTA** tab.
 - e. Ensure **Transaction Log Store** in JDBC is selected and change the **Transaction Log Prefix** name to `TLOG_WLS_XYZn`.
 - f. The rest of the fields are carried over from the server copied (including the Datasource used for the JDBC store) `WLSRuntimeSchemaDataSource`.
 - g. Click **Save** and **Commit changes** in the **Shopping Cart**.

Use the following table to identify the clusters that use JDBC TLOGs by default:

Table 21-2 The Name of Clusters that Use JDBC TLOGs by Default

Cluster to Scale Out	New Server Name	TLOG Persistent Store
WSM-PM_Cluster	WLS_WSM3	Default (file)
SOA_Cluster	WLS_SOA3	JDBC

Table 21-2 (Cont.) The Name of Clusters that Use JDBC TLOGs by Default

Cluster to Scale Out	New Server Name	TLOG Persistent Store
ESS_Cluster	WLS_ESS3	Default (file)
OSB_Cluster	WLS_OSB3	JDBC
BAM_Cluster	WLS_BAM3	JDBC
MFT_Cluster	WLS_MFT3	JDBC

11. If the cluster that you are scaling out is configured for automatic service migration, update the **JTA Migration Policy** to the required value.
 - a. Go to **Environment** and expand **Servers**. From the list of servers, select **WLS_XYZn**, click the **JTA Migratable Target**.
 - b. Use [Table 21-3](#) to ensure the recommended JTA Migration Policy depending on the cluster that you want to scale out is set.

Table 21-3 The Recommended JTA Migration Policy for the Cluster to be Scaled Out

Cluster to Scale Out	New Server Name	JTA Migration Policy
WSM-PM_Cluster	WLS_WSM3	Manual
SOA_Cluster	WLS_SOA3	Failure Recovery
ESS_Cluster	WLS_ESS3	Manual
OSB_Cluster	WLS_OSB3	Failure Recovery
BAM_Cluster	WLS_BAM3	Failure Recovery
MFT_Cluster	WLS_MFT3	Failure Recovery

- c. In the servers already existing in the cluster, verify that the list of the JTA candidate servers for JTA migration is empty:
 - i. Click **Environment** and expand **Servers**.
 - ii. Select the server.
 - iii. Select the **JTA Migratable Target** in the context menu.
 - iv. Check the **Constrained Candidate Servers** list and verify that the list is empty (an empty list indicates that all the servers in the cluster are JTA candidate servers). The list should be empty out-of-the-box so no changes are needed.
 - v. If the server list is not empty, you should modify the list to make it blank. Or, if your list is not empty because you explicitly decided to constrain the migration to some specific servers only, modify it as per your preferences to accommodate the new server. Click **Save** and **Commit Changes** in the **Shopping Cart**. Note that a change in the candidate list requires a restart of the existing servers in the cluster.
12. If the cluster you are scaling out is configured for automatic service migration, use the Oracle WebLogic Remote Console to update the automatically created WLS_XYZn (migratable) target with the recommended migration policy, because by default it is set to **Manual Service Migration Only**.

Use the following table for the list of migratable targets to update:

Table 21-4 The Recommended Migratable Targets to Update

Cluster to Scale Out	Migratable Target to Update	Migration Policy
WSM-PM_Cluster	NA	NA
SOA_Cluster	WLS_SOA3 (migratable)	Failure Recovery
ESS_Cluster	NA	NA
OSB_Cluster	WLS_OS3 (migratable)	Failure Recovery
BAM_Cluster	WLS_BAM3 (migratable)	Exactly-Once
MFT_Cluster	WLS_MFT3 (migratable)	Failure Recovery

- a. Go to **Environment** then **Migratable Targets**.
 - b. Click WLS_XYZ3 (migratable).
 - c. Change the **Service Migration Policy** to the value listed in the table.
 - d. Leave the **Constrained Candidate Server** list blank in case there are chosen servers. If no servers are selected, you can migrate this migratable target to any server in the cluster.
 - e. Click **Save** and **Commit Changes** in the **Shopping Cart**. Notice that a change from the default migration policy (manual) requires a restart of the existing servers in the cluster.
13. For components that use multiple migratable targets, in addition to step 11, Oracle WebLogic Server Remote Console create a new (migratable) target copying the settings from the existing ones in the cluster. Use the steps above for the required customizable settings.
 14. Verify that the **Constrained Candidate Server** list in the existing migratable servers in the cluster is empty. It should be empty out-of-the-box because the Configuration Wizard leaves it empty. An empty candidate list means that all the servers in the cluster are candidates, which is the best practice.
 - a. Go to each migratable server.
 - b. Click the **Migration** tab and check the **Constrained Candidate Servers** list.
 - c. Ensure that "Chosen" server list is empty. It should be empty out-of-the-box.
 - d. If the server list is not empty, you should modify the list to make it blank. Or, if your list is not empty because you explicitly decided to constrain the migration to some specific servers only, modify it as per your preferences to accommodate the new server. Click **Save** and **Commit Changes** in the **Shopping Cart**. Notice that a change in the candidate list requires a restart of the existing servers in the cluster.
 15. Create the required persistent stores for the JMS servers used in the new server.
 - a. Log into the WebLogic Remote Console. In the **Edit Tree** expand **Services** and select **JDBC stores**.
 - b. Click **New**.

Use the following table to create the required persistent stores:

 **Note:**

The number in names and prefixes in the existing resources were assigned automatically by the Configuration Wizard during the domain creation. For example:

- UMSJMSJDBCStore_auto_1 — soa_1
- UMSJMSJDBCStore_auto_2 — soa_2
- BPMJMSJDBCStore_auto_1 — soa_3
- BPMJMSJDBCStore_auto_2 — soa_4
- SOAJMSJDBCStore_auto_1 — soa_5
- SOAJMSJDBCStore_auto_2 — soa_6

So review the existing prefixes and select a new and unique prefix and name for each new persistent store.

To avoid naming conflicts and simplify the configuration, new resources are qualified with the *scaled* tag and are shown here as an example.

Table 21-5 The New Resources Qualified with the Scaled Tag

Cluster to Scale Out	Persistent Store	Prefix Name	Data Source	Target
WSM-PM_Cluster	NA	NA	NA	NA
SOA_Cluster	UMSJMSJDBCStore_soa_scaled_3	soaums_scaled_3	WLSRuntimeSchemaDataSource	WLS_SOA3 (migratable)
	SOAJMSJDBCStore_soa_scaled_3	soajms_scaled_3	WLSRuntimeSchemaDataSource	WLS_SOA3 (migratable)
	BPMJMSJDBCStore_soa_scaled_3	soabpm_scaled_3	WLSRuntimeSchemaDataSource	WLS_SOA3 (migratable)
ESS_Cluster	NA	NA	NA	NA
OSB_Cluster	UMSJMSJDBCStore_osb_scaled_3	osbums_scaled_3	WLSRuntimeSchemaDataSource	WLS_OS3 (migratable)
	OSBJMSJDBCStore_osb_scaled_3	osbjms_scaled_3	WLSRuntimeSchemaDataSource	WLS_OS3 (migratable)
BAM_Cluster	UMSJMSJDBCStore_bam_scaled_3	bamums_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamPersistenceJmsJDBCStore_bam_scaled_3	bamP_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamReportCacheJmsJDBCStore_bam_scaled_3	bamR_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamAlertEngineJmsJDBCStore_bam_scaled_3	bamA_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)

Table 21-5 (Cont.) The New Resources Qualified with the Scaled Tag

Cluster to Scale Out	Persistent Store	Prefix Name	Data Source	Target
	BamJmsJDBCStore_bam_scaled_3	bamjms_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamCQServiceJmsServersJDBCStore_bam_scaled_3	bamC_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3*
MFT_Cluster	MFTJMSJDBCStore_mft_scaled_3	mftjms_scaled_3	WLSRuntimeSchemaDataSource	WLS_MFT3 (migratable)

 **Note:**

(*) BamCQServiceJmsServers host local queues for the BAM CQService (Continuous Query Engine) and are meant to be local. They are intentionally targeted to the WebLogic servers directly and not to the migratable targets.

16. Create the required JMS Servers for the new managed server.
 - a. Go to **WebLogic Remote Console**. In the Edit Tree, select **Services**, and then click **JMS Servers**.
 - b. Click **New**.

Use the following table to create the required JMS Servers. Assign to each JMS Server the previously created persistent stores:

 **Note:**

The number in the names of the existing resources are assigned automatically by the Configuration Wizard during domain creation.

So review the existing JMS server names and select a new and unique name for each new JMS server.

To avoid naming conflicts and simplify the configuration, new resources are qualified with the *product_scaled_N* tag and are shown here as an example.

Cluster to Scale Out	JMS Server Name	Persistent Store	Target
WSM-PM_Cluster	NA	NA	NA
SOA_Cluster	UMSJMSJMServer_soa_scaled_3	UMSJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	SOAJMSJMServer_soa_scaled_3	SOAJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	BPMJMSServer_soa_scaled_3	BPMJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
ESS_Cluster	NA	NA	NA
OSB_Cluster	UMSJMSJMServer_osb_scaled_3	UMSJMSJDBCStore_osb_scaled_3	WLS_OS3 (migratable)

Cluster to Scale Out	JMS Server Name	Persistent Store	Target
	wlsbJMSServer_osb_scaled_3	OSBJMSJDBCStore_osb_scaled_3	WLS_OSB3 (migratable)
BAM_Cluster	UMSJMServer_bam_scaled_3	UMSJMSJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamPersistenceJmsServer_bam_scaled_3	BamPersistenceJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamReportCacheJmsServer_bam_scaled_3	BamReportCacheJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamAlertEngineJmsServer_bam_scaled_3	BamAlertEngineJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BAMJMSServer_bam_scaled_3	BamJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamCQServiceJmsServer_bam_scaled_3	BamCQServiceJmsJDBCStore_bam_scaled_3	WLS_BAM3*
MFT_Cluster	MFTJMSServer_mft_scaled_3	MFTJMSJDBCStore_mft_scaled_3	WLS_MFT3 (migratable)

 **Note:**

(*) BamCQServiceJmsServers host local queues for the BAM CQService (Continuous Query Engine) and are meant to be local. They are intentionally targeted to the WebLogic servers directly and not to the migratable targets.

17. Update the SubDeployment Targets for JMS Modules (if applicable) to include the recently created JMS servers.
 - a. Expand **Services** and select **JMS Modules**.
 - b. Click the JMS module. For example: `BPMJMSModule`.

Expand the **Sub Deployments** and select the corresponding one to update the targets, use the following table to identify the JMS modules to update, depending on the cluster that you are scaling out:

Table 21-6 The JMS Modules to Update

Cluster to Scale Out	JMS Module to Update	JMS Server to Add to the Subdeployment
WSM-PM_Cluster	NA	NA
SOA_Cluster	UMSJMSSystemResource *	UMSJMSServer_soa_scaled_3
	SOAJMSModule	SOAJMSServer_soa_scaled_3
	BPMJMSModule	BPMJMSServer_soa_scaled_3

Table 21-6 (Cont.) The JMS Modules to Update

Cluster to Scale Out	JMS Module to Update	JMS Server to Add to the Subdeployment
ESS_Cluster	NA	NA
OSB_Cluster	UMSJMSSystemResource *	UMSJMSServer_osb_scaled_3
	jmsResources (scope Global)	wlsbJMSServer_osb_scaled_3
BAM_Cluster	BamPersistenceJmsSystemModule	BamPersistenceJmsServer_bam_scaled_3
	BamReportCacheJmsSystemModule	BamReportCacheJmsServer_bam_scaled_3
	BamAlertEngineJmsSystemModule	BamAlertEngineJmsServer_bam_scaled_3
	BAMJMSSystemResource	BAMJMSServer_bam_scaled_3
	BamCQServiceJmsSystemModule	N/A (do not update existing subdeployments. New subdeployment for the new server will be created in next steps)
	UMSJMSSystemResource *	UMSJMSServer_bam_scaled_3
MFT_Cluster	MFTJMSModule	MFTJMSServer_mft_scaled_3

(*) Some modules (UMSJMSSystemResource) may be targeted to more than one cluster. Ensure that you update the appropriate subdeployment in each case.

- c. Add the corresponding JMS Server to the existing subdeployment.

 **Note:**

The subdeployment module name is a random name in the form of SOAJMServerXXXXXX, UMSJMSServerXXXXXX, or BPMJMSServerXXXXXX, resulting from the Configuration Wizard JMS configuration for the first two servers (WLS_SOA1 and WLS_SOA2).

- d. Click **Save** and **Commit Changes** in the **Shopping Cart**.
18. In case you are scaling out a BAM cluster, you need to create some additional resources (subdeployment and local queues) for the new server in the `BamCQServiceJmsSystemModule` module. Follow these steps to create them:
 - a. Go to **WebLogic Remote Console**. Click the **Edit Tree** and select **Environment > Services**
 - b. Click **JMS Modules** and select **BamCQServiceJmsSystemModule**.
 - c. Click **Targets**.
 - d. Add WLS_BAM3 to the targets and click **Save**.
 - e. Create a new Subdeployment in the `BamCQServiceJmsSystemModule` JMS Module with the name `BamCQServiceAlertEngineSubdeployment_scaled_3`. Then select `BamCQServiceJmsServer_bam_scaled_3` as the target of this subdeployment.

Table 21-7 Information to Create the Additional Subdeployment for Local Queues

Subdeployment Name	Subdeployment Target
BamCQServiceAlertEngineSubdeployment_scaled_3	BamCQServiceJmsServer_bam_scaled_3

- f. Select **Queues** under the **Module** and click **New**.
- g. Click **Create**.
- h. Name it `BamCQServiceAlertEngineQueue_auto_3`.
- i. Click in the newly created queue `BamCQServiceAlertEngineQueue_auto_3`
- j. Select **General** tab.
- k. Set Local JNDI Name to `queue/oracle.beam.cqservice.mdb.alertengine`.
- l. Set **Sub Deployment Name** to `BamCQServiceAlertEngineSubdeployment_scaled_3`.
- m. Click **Save** and **Commit changes** in the **Shopping Cart**.
- n. Repeat these steps to create the other queue `BamCQServiceReportCacheQueue_auto_3` with the information in [Table 21-8](#).
- o. After you finish, you have these new local queues.

Table 21-8 Information to Create the Local Queues

Name	Type	Local JNDI Name	Subdeployment
BamCQServiceAlertEngineQueue_auto_3	Queue	queue/ oracle.beam.cqservice .mdb.alertengine	BamCQServiceAlertEngineSubdeployment_scaled_3
BamCQServiceReportCacheQueue_auto_3	Queue	queue/ oracle.beam.cqservice .mdb.reportcache	BamCQServiceAlertEngineSubdeployment_scaled_3

19. The configuration is finished. Now sign in to `SOAHOST1` and run the `pack` command to create a template pack, as follows:

```
cd $ORACLE_COMMON_HOME/common/bin
./pack.sh -managed=true
        -domain=ASERVER_HOME
        -template=/full_path/scaleout_domain.jar
        -template_name=scaleout_domain_template
        tmp/pack.log
```

In this example:

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

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

```
SHARED_CONFIG_DIR/domains/template_filename.jar
```

- `scaleout_domain.jar` is a sample name for the jar file that you are creating, which contains the domain configuration files.
 - `scaleout_domain_template` is the label that is assigned to the template data stored in the template file.
20. Run the `unpack` command on `SOAHOSTN` to unpack the template in the managed server domain directory, as follows:

```
cd $ORACLE_COMMON_HOME/common/bin
./unpack.sh -domain=MSERVER_HOME
            -overwrite_domain=true
            -template=/full_path/scaleout_domain.jar
            -log_priority=DEBUG
            -log=/tmp/unpack.log
            -app_dir=APPLICATION_HOME
```

In this example:

- Replace `MSERVER_HOME` with the complete path to the domain home to be created on the local storage disk. This is the location where the copy of the domain is unpacked.
 - Replace `/full_path/scaleout_domain.jar` with the complete path and file name of the domain template jar file that you created when you ran the `pack` command to pack up the domain on the shared storage device
 - Replace `APPLICATION_HOME` with the complete path to the Application directory for the domain on shared storage. See [File System and Directory Variables Used in This Guide](#).
21. When scaling out `OSB_Cluster`:
- Restart the Admin Server to see the new server in the Service Bus Dashboard.
22. When scaling out `MFT_Cluster`:
- Default SFTP/FTP ports are used in the new server. If you are not using the defaults, configure the ports in the SFTP server as described in [Configuring the SFTP Ports](#) to configure the ports in the SFTP server.
23. Start Node Manager on the new host.

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

24. Start the new managed server.

25. Update the web tier configuration to include the new server:

- If you are using OHS, there is no need to add the new server to OHS. By default, the Dynamic Server List is used, which means that the list of servers in the cluster is automatically updated when a new node becomes part of the cluster. So, adding it to the list is not mandatory. The `WebLogicCluster` directive needs only a sufficient

number of redundant `server:port` combinations to guarantee the initial contact in case of a partial outage.

If there are expected scenarios where the Oracle HTTP Server is restarted and only the new server is up, update the `WebLogicCluster` directive to include the new server.

For example:

```
<Location /soa-infra>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004,SOAHOST3:7004
</Location>
```

Verifying the Scale Out

After scaling out and starting the server, proceed with the following verifications:

1. Verify the correct routing to web applications.

For example:

- a. Access the application on the load balancer:

```
https://soa.example.com/soa-infra
```

- b. Check that there is activity in the new server also:

In the **Remote Console**, go to **Monitoring Tree** and navigate to **Deployments > Application Runtime Data > soa-infra**.

- c. You can also verify that the web sessions are created in the new server:
 - In **Remote Console**, go to **Monitoring Tree** and navigate to **Deployments > Application Runtime Data > soa-infra**.
 - Go to **Component Runtimes** and click **<WLS_SOA3_/soa-infra**.
 - Verify if there are sessions.

You can use the sample URLs and the corresponding web applications that are identified in the following table, to check if the sessions are created in the new server for the cluster that you are scaling out:

Cluster to Verify	Sample URL to Test	Web Application Module
WSM-PM_Cluster	https:// soainternal.example.com :444/wsm-pm	wsm-pm > wsm-pm
SOA_Cluster	https:// soa.example.com/soa- infra	soa-infra > soa-infra
ESS_Cluster	https:// soa.example.com/ ESSHealthCheck	ESSHealthCheck
OSB_Cluster	https:// osb.example.com/ sbinspection.wsil	Service Bus WSIL

Cluster to Verify	Sample URL to Test	Web Application Module
MFT_Cluster	https:// mft.example.com/ mftconsole	mftconsole
BAM_Cluster	https:// soa.example.com/bam/ composer	BamComposer > /bam/ composer

2. Verify that JMS messages are being produced and consumed to the destinations, and produced and consumed from the destinations, in the three servers.
 - a. In the **Remote Console**, go to the **Monitoring Tree**.
 - b. Navigate to **Dashboards > JMS Destinations**.
3. Verify the service migration, as described in [Validating Automatic Service Migration](#).

Scaling in the Topology

This section describes how to scale in the topology for a cluster.

Perform the following steps to scale in the topology for a cluster:

1. To scale in the cluster without any JMS data loss, perform the steps described in [Managing the JMS Messages in a SOA Server](#):
 - To drain the messages, see [Draining the JMS Messages from a SOA Server](#).
 - To import the messages into another member of the cluster, see [Importing the JMS Messages into a SOA Server](#).

After you complete the steps, continue with the scale-in procedure.

2. Check the pending JTA. Before you shut down the server, review if there are any active JTA transactions in the server that you want to delete. Navigate to the WebLogic Remote Console in the monitoring tree, click **Servers > <server name> > Services > Transactions > JTA Runtime** and select the **Transactions** tab.

Note:

If you have used the **Shutdown Recovery** policy for JTA, the transactions are recovered in another server after you shut down the server.

3. Shut down the server by using the **When works completes** option.

Note:

This operation can take long time if there are active HTTP sessions or long transactions in the server. For more information about graceful shutdown, see *Using Server Life Cycle Commands in Administering Server Startup and Shutdown for Oracle WebLogic Server*

4. Use the Oracle WebLogic Remote Console to delete the new server:
 - a. Click **Edit Tree**.

- b. Go to **Environment > Servers**.
- c. Select the server that you want to delete.
- d. Click **Delete**.
- e. Click **Save** and **Commit changes** in the **Shopping Cart**.

 **Note:**

If migratable target was not deleted in the previous step, you get the following error message:

```
The following failures occurred: --MigratableTargetMBean WLS_SOA3_soa-
failure-recovery (migratable) does not have a preferred server set.
Errors must be corrected before proceeding.
```

5. Use the Oracle WebLogic Remote Console to update the subdeployment of each JMS Module that is used by the cluster that you are shrinking.

Use the following table to identify the module for each cluster and perform this action for each module:

Cluster to Scale in	JMS Module	JMS Server to Delete from the Subdeployment
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	UMSJMSSystemResource SOAJMSModule BPMJMSModule	UMSJMSServer_soa_scaled_3 SOAJMSServer_soa_scaled_3 BPMJMSServer_soa_scaled_3
ESS_Cluster	Not applicable	Not applicable
OSB_Cluster	UMSJMSSystemResource jmsResources (scope Global)	UMSJMSServer_osb_scaled_3 wlsbJMSServer_osb_scaled_3
BAM_Cluster	BamPersistenceJmsSystemModule BamReportCacheJmsSystemModule BamAlertEngineJmsSystemModule BAMJMSSystemResource BamCQServiceJmsSystemModule	BamPersistenceJmsServer_bam_scaled_3 BamReportCacheJmsServer_bam_scaled_3 BamAlertEngineJmsServer_bam_scaled_3 BAMJMSServer_bam_scaled_3 Not applicable (existing subdeployments are not modified on scale-out)
MFT_Cluster	MFTJMSModule	MFTJMSServer_mft_scaled_3

- a. Click **Edit Tree**.
- b. Go to **Services > JMS System Resources**.
- c. Click the JMS module.
- d. Click **Sub Deployment**.
- e. Select the **Sub Deployment** Module
- f. Unselect the JMS server that was created for the deleted server.

- g. Click **Save** and **Commit changes** in the **Shopping Cart**.
6. In case you want to scale in a BAM cluster, use the Oracle WebLogic Remote Console to delete the local queues that are created for the new server:
 - a. Click **Edit Tree**.
 - b. Go to **Services>JMS Modules**.
 - c. Click the JMS module.
 - d. Click in `BamCQServiceJmsSystemModule`.
 - e. Delete the local queues that are created for the new server:
 - `BamCQServiceAlertEngineQueue_auto_3`
 - `BamCQServiceReportCacheQueue_auto_3`
 - f. Delete the following subdeployment created for the server:
`BamCQServiceAlertEngineSubdeployment_scaled_3`
 - g. Click **Save** and **Commit changes** in the **Shopping Cart**.
7. Use the Oracle WebLogic Remote Console to delete the JMS servers:
 - a. Click **Edit Tree**.
 - b. Go to **Services > JMS Servers**.
 - c. Select the JMS Servers that you created for the new server.
 - d. Click **Delete**.
 - e. Click **Save** and **Commit changes** in the **Shopping Cart**.
8. Use the Oracle WebLogic Remote Console to delete the JMS persistent stores:
 - a. Click **Edit Tree**.
 - b. Go to **Services > JDBC Stores**.
 - c. Select the JDBC Store that you created for the new server.
 - d. Click **Delete**.
 - e. Click **Save** and **Commit changes** in the **Shopping Cart**.
9. If the machine that was hosting the deleted server is not used by any other servers you must delete it performing the following steps:
 - a. Click **Edit Tree**.
 - b. Go to **Environment > Machines**.
 - c. Select the machine that you created for the new server.
 - d. Click **Delete**.
 - e. Click **Save** and **Commit changes** in the **Shopping Cart**.
10. Update the Web tier configuration to remove references to the deleted server.

Scaling Up the Topology

This section describes how to scale up the topology.

You already have a node that runs a managed server that is configured with Fusion Middleware components. The node contains a WebLogic Server home and an Oracle Fusion Middleware SOA home in shared storage. Use these existing installations and domain

directories, to create the new managed servers. You do not need to install WLS or SOA binaries or to run *pack* and *unpack* because the new server is going to run in the existing node.

- [Prerequisites for Scaling Up](#)
- [Scaling Up](#)
- [Verifying the Scale Up of Clusters](#)

Prerequisites for Scaling Up

Before you perform a scale up of the topology, you must ensure that you meet the following requirements:

- The starting point is a cluster with managed servers already running.
- It is assumed that the cluster syntax is used for all internal RMI invocations, JMS adapter, and so on.

Scaling Up

Use the SOA EDG topology as a reference, with two application tier hosts (SOAHOST1 and SOAHOST2), each running one managed server of each cluster. The example explains how to add a third managed server to the cluster that runs in SOAHOST1. `WLS_XYZn` is the generic name given to the new managed server that you add to the cluster. Depending on the cluster that is being extended and the number of existing nodes, the actual names are `WLS_SOA3`, `WLS_OSB3`, `WLS_ESS3`, and so on.

The scale-up procedure requires downtime for the existing servers in the WLS cluster being scaled if service migration has been configured for them with a different migration policy from the default one (manual). It also implies downtime if the existing migratable targets do not use an empty Candidate Server list (a precise subset of servers in the cluster is used as candidates). Using empty candidate lists is the best practice because it means that all the servers in the cluster are candidates for migration. You can check the list of candidates for each migratable targets through the Weblogic Remote Console:

1. Access the domain with the WebLogic Remote Console.
2. Click the **Edit Tree** at the top left side in the Remote Console.
3. Expand **Environment** in the navigation tree on the left.
4. Expand **Migratable targets** in the navigation tree on the left.
5. Click each migratable target and verify the **Constrained Candidate Servers** list under **Migration** tab.

If you have created your environment following the Enterprise Deployment Guide, these lists are empty out-of-the-box. When you add a new server to the cluster, the server is automatically considered for migration without the need to restart the existing servers.

If you had decided to constraint the migration to some specific servers of the cluster only, your Candidate Server lists will not be empty. When you add a new server to the cluster, you may need to modify them to add the new server. In this case, you will have to restart the existing nodes during the scale-out process. Changing migration policy from the manual one for the new server also prompts for a restart of existing members in the cluster. Oracle recommends that you “batch” these two changes and perform one single restart after you complete both these changes (migration policy and list of candidates).

To scale up the cluster, complete the following steps:

1. Use the Oracle WebLogic Remote Console to clone the first managed server in the cluster into a new managed server.
 - a. Go to **Environment** and select **Servers**.
 - b. Click **Create**, in the **Copy settings from another server** select the first managed server in the cluster to scale out and click **Create**.
 - c. Use [Table 21-1](#) to set the correspondent name, listen address, and SSL listen port depending on the cluster that you want to scale out.

 **Note:**

The port value is incremented by 1 to avoid binding conflicts with the managed server that is already created and running in the same host.

- d. Click the new managed server, select **Configuration**, and then click **General**.
- e. Verify that the Machine assigned is SOAHOST1.
- f. Update the Administration port for the server to be consistent with other server in the cluster. Note that the port value is incremented by 1 to avoid binding conflicts with the managed server that is already created and running in the same host.

Table 21-9 List of Clusters that You Want to Scale Up

Cluster to Scale Up	Server to Clone	New Server Name	Server Listen Address	SSL Server Listen Port	Local Administration Port Override Scale up
WSM-PM_Cluster	WLS_WSM1	WLS_WSM3	SOAHOST1	7011	9013
SOA_Cluster	WLS_SOA1	WLS_SOA3	SOAHOST1	7005	9024
ESS_Cluster	WLS_ESS1	WLS_ESS3	SOAHOST1	7009	9016
OSB_Cluster	WLS_OSB1	WLS_OSB3	SOAHOST1	8004	9017
BAM_Cluster	WLS_BAM1	WLS_BAM3	SOAHOST1	7007	9015
MFT_Cluster	WLS_MFT1	WLS_MFT3	MFTHOST1	7011	9024

2. Update the deployment Staging Directory Name of the new server, as described in [Modifying the Upload and Stage Directories to an Absolute Path in an Enterprise Deployment](#).
3. Your new server's keystore location and ssl configuration is carried over from the server copied (WLS_SOA1) but it is required to update the password again (since it will be encrypted again for the new server) and the "Server private key alias" entry for this new server.
 - a. Navigate to **Environment > Servers**.
 - b. Click on the new server.
 - c. Navigate to **Security > Keystores**.
 - d. Update the **Custom Identity Key Store Pass Phrase** and **Custom Trust Key Store Pass Phrase** with the password provided to the `generate_perdomainCACERTS.sh` script.

- e. Click on the **SSL** tab under **Security**.
 - f. Update the **Server Private Key Pass Phrase** with the password provided to the `generate_perdomainCACERTS.sh` script
 - g. Click **Save** and **Commit changes** in the **Shopping Cart**.
4. Update the TLOG JDBC persistent store of the new managed server:
- a. Log into the WebLogic Remote Console.
 - b. Go to **Environment** and expand the **Servers** link on the navigation tree on the left.
 - c. Click the new server **WLS_XYZn**.
 - d. Click the **Services > JTA** tab.
 - e. Ensure **Transaction Log Store** in JDBC is selected and change the **Transaction Log Prefix** name to **TLOG_WLS_XYZn**.

The rest of the fields are carried over from the server copied (including the Datasource used for the JDBC store).
 - f. Click **Save** and **Commit changes** in the **Shopping Cart**.

Use the following table to identify the clusters that use JDBC TLOGs by default:

Table 21-10 The Name of Clusters that Use JDBC TLOGs by Default

Cluster to Scale Up	New Server Name	TLOG Persistent Store
WSM-PM_Cluster	WLS_WSM3	Default (file)
SOA_Cluster	WLS_SOA3	JDBC
ESS_Cluster	WLS_ESS3	Default (file)
OSB_Cluster	WLS_OSB3	JDBC
BAM_Cluster	WLS_BAM3	JDBC
MFT_Cluster	WLS_MFT3	JDBC

5. If the cluster you are scaling up is configured for automatic service migration, update the **JTA Migration Policy** to the required value.

Use the following table to identify the clusters for which you have to update the JTA Migration Policy:

Table 21-11 The Recommended JTA Migration Policy for the Cluster to be Scaled Up

Cluster to Scale Up	New Server Name	JTA Migration Policy
WSM-PM_Cluster	WLS_WSM3	Manual
SOA_Cluster	WLS_SOA3	failure-recovery
ESS_Cluster	WLS_ESS3	Manual
OSB_Cluster	WLS_OSB3	failure-recovery
BAM_Cluster	WLS_BAM3	failure-recovery
MFT_Cluster	WLS_MFT3	failure-recovery

Complete the following steps:

- a. Go to **Environment Tree** and select **Servers**. From the list of servers, select **WLS_XYZn**, click **JTA Migratable**.

- b. Use [Table 21-11](#) to set the recommended JTA Migration Policy depending on the cluster that you want to scale out.
- c. Click **Save** and **Commit changes** in the **Shopping Cart**.
- d. In the servers already existing in the cluster, verify that the list of the JTA candidate servers for JTA migration is empty:
 - i. Click **Environment** and expand **Servers**.
 - ii. Select the server.
 - iii. Select the **JTA Migratable Target** in the context menu.
 - iv. Check the **Constrained Candidate Servers** list and verify that the list is empty (an empty list indicates that all the servers in the cluster are JTA candidate servers). The list should be empty out-of-the-box so no changes are needed.
 - v. If the server list is not empty, you should modify the list to make it blank. Or, if your list is not empty because you explicitly decided to constrain the migration to some specific servers only, modify it as per your preferences to accommodate the new server. Save and commit the changes. Restart the existing servers for this change to become effective.
6. If the cluster you are scaling up is configured for automatic service migration, use the Oracle WebLogic Remote Console to update the automatically created WLS_XYZn (migratable) with the recommended migration policy, because by default it is set to **Manual Service Migration Only**.

Use the following table for the list of migratable targets to update:

Table 21-12 The Recommended Migratable Targets to Update

Cluster to Scale Up	Migratable Target to Update	Migration Policy
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	WLS_SOA3 (migratable)	failure-recovery
ESS_Cluster	Not applicable	Not applicable
OSB_Cluster	WLS_OSB3 (migratable)	failure-recovery
BAM_Cluster	WLS_BAM3 (migratable)	exactly-once
MFT_Cluster	WLS_MFT3 (migratable)	failure-recovery

- a. Go to **Environment > Migratable Targets**.
- b. Click **WLS_XYZ3 (migratable)**.
- c. Change the Service Migration Policy to the value listed in the table.
- d. Leave the Constrained Candidate Server list blank in case there are chosen servers. If no servers are selected, you can migrate this migratable target to any server in the cluster.
- e. Click **Save** and **Commit changes** in the **Shopping Cart**. Notice that a change from the default migration policy (manual) requires restart.
7. For components that use multiple migratable targets in addition to *Step 11*, Oracle WebLogic Server Remote Console create a new migratable target copying the settings from the existing ones in the cluster. Use the steps above for the required customizable settings.
8. Verify that the **Constrained Candidate Server** list in the existing migratable servers in the cluster is empty. It should be empty out-of-the-box because the Configuration Wizard

leaves it empty. An empty candidate list means that all the servers in the cluster are candidates, which is the best practice.

- a. Go to each migratable server.
 - b. Click the **Migration** tab and check the **Constrained Candidate Servers** list.
 - c. Ensure that Chosen server list is empty. It should be empty out-of-the-box.
 - d. If the server list is not empty, you should modify the list to make it blank. Or, if your list is not empty because you explicitly decided to constraint the migration to some specific servers only, modify it as per your preferences to accommodate the new server. Click **Save** and **Commit Changes** in the **Shopping Cart**. Restart the existing servers for this change to become effective
9. Create the required persistent stores for the JMS servers.
- a. Sign into the WebLogic Remote Console and go to **Services** and select **JDBC Stores**.
 - b. Click **New** and select **Create JDBCStore**.

Use the following table to create the required persistent stores:

 **Note:**

The number in the names and prefixes in the existing resources were assigned automatically by the Configuration Wizard during the domain creation.

For example:

```
UMSJMSJDBCStore_auto_1 - soa_1
UMSJMSJDBCStore_auto_2 - soa_2
BPMJMSJDBCStore_auto_1 - soa_3
BPMJMSJDBCStore_auto_2 - soa_4
SOAJMSJDBCStore_auto_1 - soa_5
SOAJMSJDBCStore_auto_2 - soa_6
```

Review the existing prefixes and select a new and unique prefix and name for each new persistent store.

To avoid naming conflicts and simplify the configuration, new resources are qualified with the *scaled* tag and are shown here as an example.

Table 21-13 The New Resources Qualified with the Scaled Tag

Cluster to Scale Up	Persistent Store	Prefix Name	Data Source	Target
WSM-PM_Cluster	Not applicable	Not applicable	Not applicable	Not applicable
SOA_Cluster	UMSJMSJDBCStore_soa_scaled_3	soaums_scaled_3	WLSRuntimeSchemaDataSource	WLS_SOA3 (migratable)
	SOAJMSJDBCStore_soa_scaled_3	soajms_scaled_3	WLSRuntimeSchemaDataSource	WLS_SOA3 (migratable)
	BPMJMSJDBCStore_soa_scaled_3	soabpm_scaled_3	WLSRuntimeSchemaDataSource	WLS_SOA3 (migratable)
ESS_Cluster	Not applicable	Not applicable	Not applicable	Not applicable

Table 21-13 (Cont.) The New Resources Qualified with the Scaled Tag

Cluster to Scale Up	Persistent Store	Prefix Name	Data Source	Target
OSB_Cluster	UMSJMSJDBCStore_osb_scaled_3	osbums_scaled_3	WLSRuntimeSchemaDataSource	WLS_OSB3 (migratable)
	OSBJMSJDBCStore_osb_scaled_3	osbjms_scaled_3	WLSRuntimeSchemaDataSource	WLS_OSB3 (migratable)
BAM_Cluster	UMSJMSJDBCStore_bam_scaled_3	bamums_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamPersistenceJmsJDBCStore_bam_scaled_3	bamP_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamReportCacheJmsJDBCStore_bam_scaled_3	bamR_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamAlertEngineJmsJDBCStore_bam_scaled_3	bamA_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamJmsJDBCStore_bam_scaled_3	bamjms_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3_bam-exactly-once (migratable)
	BamCQServiceJmsJDBCStore_bam_scaled_3	bamC_scaled_3	WLSRuntimeSchemaDataSource	WLS_BAM3*
MFT_Cluster	MFTJMSJDBCStore_mft_scaled_3	mftjms_scaled_3	WLSRuntimeSchemaDataSource	WLS_MFT3 (migratable)

 **Note:**

(*) BamCQServiceJmsServers host local queues for the BAM CQService (Continuous Query Engine) and are meant to be local. They are intentionally targeted to the WebLogic servers directly and not to the migratable targets.

10. Create the required JMS Servers for the new managed server.
 - a. Go to **WebLogic Remote Console**. In the **Edit Tree**, select **Services**, and click **JMS Servers**.
 - b. Click **New**.

Use the following table to create the required JMS Servers. Assign to each JMS Server the previously created persistent stores:

 **Note:**

The number in the names of the existing resources are assigned automatically by the Configuration Wizard during domain creation. Review the existing JMS server names and select a new and unique name for each new JMS server. To avoid naming conflicts and simplify the configuration, new resources are qualified with the *product_scaled_N* tag and are shown here as an example.

Cluster to Scale Up	JMS Server Name	Persistent Store	Target
WSM-PM_Cluster	Not applicable	Not applicable	Not applicable
SOA_Cluster	UMSJMSSErver_soa_scaled_3	UMSJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	SOAJMSSErver_soa_scaled_3	SOAJMSJDBCStore_soa_scaled_3	WLS_SOA3 (migratable)
	BPMJMSErver_soa_scaled_3	BPMJMSErver_soa_scaled_3	WLS_SOA3 (migratable)
ESS_Cluster	Not applicable	Not applicable	Not applicable
OSB_Cluster	UMSJMSSErver_osb_scaled_3	UMSJMSJDBCStore_osb_scaled_3	WLS_OS3 (migratable)
	wlsbJMSErver_osb_scaled_3	OSBJMSJDBCStore_osb_scaled_3	WLS_OS3 (migratable)
BAM_Cluster	UMSJMSSErver_bam_scaled_3	UMSJMSJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamPersistenceJmsSErver_bam_scaled_3	BamPersistenceJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamReportCacheJmsSErver_bam_scaled_3	BamReportCacheJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamAlertEngineJmsSErver_bam_scaled_3	BamAlertEngineJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BAMJMSErver_bam_scaled_3	BamJmsJDBCStore_bam_scaled_3	WLS_BAM3_bam-exactly-once (migratable)
	BamCQServiceJmsSErver_bam_scaled_3	BamCQServiceJmsJDBCStore_bam_scaled_3	WLS_BAM3*
MFT_Cluster	MFTJMSErver_mft_scaled_3	MFTJMSErver_mft_scaled_3	WLS_MFT3 (migratable)

 **Note:**

(*) BamCQServiceJmsServers host local queues for the BAM CQService (Continuous Query Engine) and are meant to be local. They are intentionally targeted to the WebLogic servers directly and not to the migratable targets.

11. Update the SubDeployment Targets for JMS Modules (if applicable) to include the recently created JMS servers.
 - a. Expand **Services**, select **JMS Modules**, and then click the **JMS module**. For example, **BPMJMSModule**.
 - b. Expand the **Sub Deployments** and select the corresponding one to update the targets. Use the following table to identify the JMS modules to update, depending on the cluster that you are scaling out:

Use the following table to identify the JMS modules to update depending on the cluster that you are scaling up:

Cluster to Scale-up	JMS Module to Update	JMS Server to Add to the Subdeployment
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	UMSJMSSystemResource *	UMSJMSServer_soa_scaled_3
	SOAJMSModule	SOAJMSServer_soa_scaled_3
	BPMJMSModule	BPMJMSServer_soa_scaled_3
ESS_Cluster	Not applicable	Not applicable
OSB_Cluster	UMSJMSSystemResource *	UMSJMSServer_osb_scaled_3
	jmsResources (scope Global)	wlsbJMSServer_osb_scaled_3
BAM_Cluster	BamPersistenceJmsSystemModule	BamPersistenceJmsServer_bam_scaled_3
	BamReportCacheJmsSystemModule	BamReportCacheJmsServer_bam_scaled_3
	BamAlertEngineJmsSystemModule	BamAlertEngineJmsServer_bam_scaled_3
	BAMJMSSystemResource	BAMJMSServer_bam_scaled_3
	BamCQServiceJmsSystemModule	Not applicable (Do not update existing subdeployments. New subdeployment for the new server will be created in next steps)
	UMSJMSSystemResource *	UMSJMSServer_bam_scaled_3 *
MFT_Cluster	MFTJMSModule	MFTJMSServer_mft_scaled_3

(*) Some modules (UMSJMSSystemResource) may be targeted to more than one cluster. Ensure that you update the appropriate subdeployment in each case.

- c. Add the corresponding JMS Server to the existing subdeployment.

 **Note:**

The Subdeployment module name is a random name in the form of SOAJMSServerXXXXXX, UMSJMSServerXXXXXX, or BPMJMSServerXXXXXX, resulting from the Configuration Wizard JMS configuration for the first two servers (WLS_SOA1 and WLS_SOA2).

- d. Click **Save** and **Commit changes** in the **Shopping Cart**.
12. In case you are scaling out a BAM cluster, you need to create some additional resources (subdeployment and local queues) for the new server in the BamCQServiceJmsSystemModule module. Follow these steps to create them:
 - a. Go to WebLogic Remote Console, click the **Edit tree** and **Environment > Services**.
 - b. Click **Jms System Resources** and select the **BamCQServiceJmsSystemModule**.
 - c. Click **Targets**.

- d. Add `WLS_BAM3` to the targets and click **Save**.
- e. Create a new Subdeployment in the `BamCQServiceJmsSystemModule` JMS Module with the name `BamCQServiceAlertEngineSubdeployment_scaled_3`. Then select `BamCQServiceJmsServer_bam_scaled_3` as the target of this subdeployment.

Table 21-14 Information to Create the Additional Subdeployment for Local Queues

Subdeployment Name	Subdeployment Target
<code>BamCQServiceAlertEngineSubdeployment_scaled_3</code>	<code>BamCQServiceJmsServer_bam_scaled_3</code>

- f. Select **Queues** under the **Module** and click **New**.
- g. Name it `BamCQServiceAlertEngineQueue_auto_3`.
- h. Click **Create**.
- i. Click in the newly created queue `BamCQServiceAlertEngineQueue_auto_3`.
- j. Select **General** tab.
- k. Set **Local JNDI Name** to `queue/oracle.beam.cqservice.mdb.alertengine`.
- l. Set **Sub Deployment Name** to `BamCQServiceAlertEngineSubdeployment_scaled_3`.
- m. Click **Save** and **Commit changes** in the **Shopping Cart**.
- n. Repeat these steps to create the other queue `BamCQServiceReportCacheQueue_auto_3` with the information in [Table 21-15](#).
- o. After you finish, you have the following new local queues.

Table 21-15 Information to Create the Local Queues

Name	Type	Local JNDI Name	Subdeployment
<code>BamCQServiceAlertEngineQueue_auto_3</code>	Queue	<code>queue/oracle.beam.cqservice.mdb.alertengine</code>	<code>BamCQServiceAlertEngineSubdeployment_scaled_3</code>
<code>BamCQServiceReportCacheQueue_auto_3</code>	Queue	<code>queue/oracle.beam.cqservice.mdb.reportcache</code>	<code>BamCQServiceAlertEngineSubdeployment_scaled_3</code>

13. Start the new managed server.
14. When scaling up **OSB_Cluster**:
Restart the Admin Server to see the new server in the Service Bus Dashboard.
15. When scaling up the **MFT_Cluster**:
Default SFTP/FTP ports are used in the new server. If you are not using the defaults, follow the steps described in [Configuring the SFTP Ports](#) to configure the ports in the SFTP server . When scaling up, use different ports SFTP/FTP for the new server that do not conflict with the existing server in the same machine.
16. Update the web tier configuration to include this new server:
 - If you are using OHS, there is no need to add the new server to OHS. By default Dynamic Server List is used, which means that the list of the servers in the cluster is automatically updated when a new node become part of the cluster, so adding it to the

list is not mandatory. The `WebLogicCluster` directive needs only a sufficient number of redundant `server:port` combinations to guarantee initial contact in case of a partial outage.

If there are expected scenarios where the Oracle HTTP Server is restarted and only the new server would be up, update the `WebLogicCluster` directive to include the new server.

```
<Location /soa-infra>
  WLSRequest ON
  WebLogicCluster SOAHOST1:7004,SOAHOST2:7004,SOAHOST2:7005
</Location>
```

Verifying the Scale Up of Clusters

After scaling out and starting the server, proceed with the following verifications:

1. Verify the correct routing to web applications.

For example:

- a. Access the application on the load balancer:

```
https://soa.example.com/soa-infra
```

- b. Check that there is activity in the new server also:

In the **Remote Console**, go to **Monitoring Tree** and navigate to **Deployments > Application Runtime Data > soa-infra**.

- c. You can also verify that the web sessions are created in the new server:
 - In **Remote Console**, go to **Monitoring Tree** and navigate to **Deployments > Application Runtime Data > soa-infra**.
 - Go to **Component Runtimes** and click **WLS_SOA3_/soa-infra**.
 - Verify if there are sessions.

You can use the sample URLs and the corresponding web applications that are identified in the following table, to check if the sessions are created in the new server for the cluster that you are scaling out:

Cluster to Verify	Sample URL to Test	Web Application Module
WSM-PM_Cluster	https:// soainternal.example.com :444/wsm-pm	wsm-pm > wsm-pm
SOA_Cluster	https:// soa.example.com/soa- infra	soa-infra > soa-infra
ESS_Cluster	https:// soa.example.com/ ESSHealthCheck	ESSHealthCheck
OSB_Cluster	https:// osb.example.com/ sbinspection.wsil	Service Bus WSIL

Cluster to Verify	Sample URL to Test	Web Application Module
MFT_Cluster	https:// mft.example.com/ mftconsole	mftconsole
BAM_Cluster	https:// soa.example.com/bam/ composer	BamComposer > /bam/ composer

2. Verify that JMS messages are being produced and consumed to the destinations, and produced and consumed from the destinations, in the three servers.
 - a. In **Remote Console**, go to **Monitoring Tree**.
 - b. Navigate to **Dashboards > JMS Destinations**.
3. Verify the service migration, as described in [Validating Automatic Service Migration](#).

Scaling Down the Topology

This section describes how to scale down the topology.

To scale down the topology:

1. To scale in the cluster without any JMS data loss, perform the steps described in [Managing the JMS Messages in a SOA Server](#):
 - To drain the messages, see [Draining the JMS Messages from a SOA Server](#).
 - To import the messages into another member of the cluster, see [Importing the JMS Messages into a SOA Server](#).

After you complete the steps, continue with the scale-in procedure.

2. Check the pending JTA. Before you shut down the server, review if there are any active JTA transactions in the server that you want to delete. Navigate to the WebLogic Remote Console and in the **Monitoring Tree** click **Servers > <server name> > Services > Transactions > JTA Runtime**.

Note:

If you have used the **Shutdown Recovery** policy for JTA, the transactions are recovered in another server after you shut down the server.

3. Shut down the server by using the **When works completes** option.

Note:

This operation can take long time if there are active HTTP sessions or long transactions in the server. For more information about graceful shutdown, see *Using Server Life Cycle Commands in Administering Server Startup and Shutdown for Oracle WebLogic Server*

4. Use the Oracle WebLogic Server Remote Console to delete the new server:
 - a. Click **Edit Tree**.

- b. Go to **Environment > Servers**.
- c. Select the server that you want to delete.
- d. Click **Delete**.
- e. Click **Save** and **Commit changes** in the **Shopping Cart**.

 **Note:**

If migratable target was not deleted in the previous step, you get the following error message:

```
The following failures occurred: --MigratableTargetMBean
WLS_SOA3_soa-failure-recovery (migratable) does not have a
preferred server set.
Errors must be corrected before proceeding.
```

5. Use the Oracle WebLogic Server Remote Console to update the subdeployment of each JMS Module that is used by the cluster that you are shrinking.

Use the following table to identify the module for each cluster and perform this action for each module:

Table 21-16 Identify the Module for Each Cluster

Cluster to Scale in	JMS Module	JMS Server to Delete from the Subdeployment
WSM-PM_Cluster	Not applicable	Not applicable
SOA_Cluster	UMSJMSSystemResource SOAJMSModule BPMJMSModule	UMSJMSServer_soa_scaled_3 SOAJMSServer_soa_scaled_3 BPMJMSServer_soa_scaled_3
ESS_Cluster	Not applicable	Not applicable
OSB_Cluster	UMSJMSSystemResource jmsResources (scope Global)	UMSJMSServer_osb_scaled_3 wlsbJMSServer_osb_scaled_3
BAM_Cluster	BamPersistenceJmsSystemModule BamReportCacheJmsSystemModule BamAlertEngineJmsSystemModule BAMJMSSystemResource BamCQServiceJmsSystemModule	BamPersistenceJmsServer_bam_scaled_3 BamReportCacheJmsServer_bam_scaled_3 BamAlertEngineJmsServer_bam_scaled_3 BAMJMSServer_bam_scaled_3 Not applicable (existing subdeployments are not modified on scale-up)
MFT_Cluster	MFTJMSModule	MFTJMSServer_mft_scaled_3

- a. Click **Edit Tree**.
- b. Go to **Services > JMS System Resources**.
- c. Click the **JMS module**.

- d. Click **Sub Deployments**.
 - e. Select the **Sub Deployment Module**.
 - f. Unselect the JMS server that was created for the deleted server.
 - g. Click **Save** and **Commit changes** in the **Shopping Cart**.
6. In case you want to scale in a BAM cluster, use the Oracle WebLogic Remote Console to delete the local queues that are created for the new server:
- a. Click **Edit Tree**.
 - b. Go to **Services > JMS Modules**.
 - c. Click the **JMS module**.
 - d. Click `BamCQServiceJmsSystemModule`.
 - e. Delete the local queues that are created for the new server:


```
BamCQServiceAlertEngineQueue_auto_3
```

```
BamCQServiceReportCacheQueue_auto_3
```
 - f. Delete the subdeployment created for the server:


```
BamCQServiceAlertEngineSubdeployment_scaled_3
```
 - g. Click **Save** and **Commit changes** in the **Shopping Cart**.
7. Use the Oracle WebLogic Remote Console to delete the JMS servers:
- a. Click **Edit Tree**.
 - b. Go to **Services > JMS Servers**.
 - c. Select the JMS Servers that you created for the new server.
 - d. Click **Delete**.
 - e. Click **Save** and **Commit changes** in the **Shopping Cart**.
8. Use the Oracle WebLogic Server Remote Console to delete the JMS persistent stores:
- a. Click **Edit Tree**.
 - b. Go to **Services > JDBC Stores**.
 - c. Select the JDBC Store that you created for the new server.
 - d. Click **Delete**.
 - e. Click **Save** and **Commit changes** in the **Shopping Cart**.
9. Update the web tier configuration to remove references to the deleted server.
10. If the Machine that was hosting the deleted server is not used by any other servers you can also delete it.
- a. Click **Edit Tree**.
 - b. Go to **Environment > Machines**.
 - c. Select the Machine that you created for the new server.
 - d. Click **Delete**.
 - e. Click **Save** and **Commit changes** in the **Shopping Cart**.

A

Targeting Applications and Resources to Servers

The component-wise list of targets is used to verify that the value used in the `config.xml` file is correct.

This appendix lists the applications, library, startup class, shutdown class, JMS system resource, and JDBC system resource targets for an Oracle SOA enterprise deployment.

- [Oracle SOA Enterprise Application Targets](#)
This section provides a table that lists the Oracle SOA enterprise deployment application targets.
- [Oracle SOA Enterprise Deployment Library Targets](#)
This section provides a table that lists the Oracle SOA enterprise deployment library targets.
- [Oracle SOA Enterprise Deployment Startup Class Targets](#)
This section provides a table that lists the Oracle SOA enterprise deployment Startup Class targets.
- [Oracle SOA Enterprise Deployment Shutdown Class Targets](#)
This section provides a table that lists the Oracle SOA enterprise deployment Shutdown Class targets.
- [Oracle SOA Enterprise Deployment JMS System Resource Targets](#)
This section provides a table that lists the Oracle SOA enterprise deployment JMS System Resource targets.
- [Oracle SOA Enterprise Deployment JDBC System Resource Targets](#)
This section provides a table that lists the Oracle SOA enterprise deployment JDBC System Resource targets.

Oracle SOA Enterprise Application Targets

This section provides a table that lists the Oracle SOA enterprise deployment application targets.

These are the default targets resulting from the EDG configuration. Other customizations (for example, extending the File Adapter in OSB) may require additional targeting.

Table A-1 SOA Application Targets

Application	Targets
api-console	SOA_Cluster (BPM)
Aggregator Singleton Marker Application	OSB_Cluster
AqAdapter	OSB_Cluster, AdminServer, SOA_Cluster, MFT_Cluster
b2bui	SOA_Cluster
BamComposer	BAM_Cluster

Table A-1 (Cont.) SOA Application Targets

Application	Targets
BamCQService	BAM_Cluster
BamServer	BAM_Cluster
BPMComposer	SOA_Cluster (BPM)
Cloudsdk	AdminServer, OSB_Cluster, SOA_Cluster
coherence-transaction-rar	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
CoherenceAdapter	SOA_Cluster
DbAdapter	AdminServer, OSB_Cluster, SOA_Cluster, MFT_Cluster
DefaultToDoTaskFlow	SOA_Cluster
DMS Application#12.2.1.1.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
em	AdminServer
ESSAPP	ESS_Cluster
ESSNativeHostingApp#V1.0	ESS_Cluster
FileAdapter	AdminServer, OSB_Cluster, SOA_Cluster, MFT_Cluster
FtpAdapter	AdminServer, OSB_Cluster, SOA_Cluster, MFT_Cluster
JDEWorldAdapter	SOA_Cluster
JMSAdapter	AdminServer, OSB_Cluster, SOA_Cluster, MFT_Cluster
LdapAdapter	SOA_Cluster
MQSeriesAdapter	SOA_Cluster
MSMQAdapter	SOA_Cluster
OAAPredictionService	SOA_Cluster (BPM)
opss-rest	AdminServer, BAM_Cluster
OracleAppsAdapter	SOA_Cluster
OracleBPMBACServerApp	SOA_Cluster (BPM)
OracleBPMComposerRolesApp	SOA_Cluster (BPM)
OracleBPMProcessRolesApp	SOA_Cluster (BPM)
OracleBPMWorkspace	SOA_Cluster (BPM)
OracleBamAdapter	
SAPAdapter	
Service Bus Domain Singleton Marker Application	
Service Bus DSP Transport Provider	AdminServer, OSB_Cluster
Service Bus EJB Transport Provider	AdminServer, OSB_Cluster
Service Bus Email Transport Provider	AdminServer, OSB_Cluster

Table A-1 (Cont.) SOA Application Targets

Application	Targets
Service Bus File Transport Provider	AdminServer, OSB_Cluster
Service Bus Framework Starter Application	AdminServer, OSB_Cluster
Service Bus FTP Transport Provider	AdminServer, OSB_Cluster
Service Bus JCA Transport Provider	AdminServer, OSB_Cluster
Service Bus JEJB Transport Provider	AdminServer, OSB_Cluster
Service Bus JMS Reporting Provider	OSB_Cluster
Service Bus Kernel	AdminServer, OSB_Cluster
Service Bus Logging	AdminServer, OSB_Cluster
Service Bus Message Reporting Purger	OSB_Cluster
Service Bus MQ Transport Provider	AdminServer, OSB_Cluster
Service Bus OWSM Initializer	AdminServer, OSB_Cluster
Service Bus Publish	AdminServer, OSB_Cluster
Service Bus REST Deployment	AdminServer
Service Bus Resource	OSB_Cluster
Service Bus Result Cache	OSB_Cluster
Service Bus Routing	AdminServer, OSB_Cluster
Service Bus SB Transport Provider	AdminServer, OSB_Cluster
Service Bus SFTP Transport Provider	AdminServer, OSB_Cluster
Service Bus SOA-DIRECT Transport Provider	AdminServer, OSB_Cluster
Service Bus Subscription Listener	OSB_Cluster
Service Bus TEST_Console	AdminServer
Service Bus Test Framework	AdminServer, OSB_Cluster
Service Bus Transform	AdminServer, OSB_Cluster
Service Bus Tuxedo Transport Provider	AdminServer, OSB_Cluster
Service Bus UDDI Manager	AdminServer
Service Bus WS Transport Async Response	OSB_Cluster
Service Bus WS Transport Provider	OSB_Cluster
Service Bus WSIL	OSB_Cluster
service-bus	AdminServer
SimpleApprovalTaskFlow	SOA_Cluster
SiebelAdapter	SOA_Cluster
soa-infra	SOA_Cluster
soa-webapps	SOA_Cluster
SocketAdapter	SOA_Cluster
state-management-provider-memory-rar	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
UMSAdapter	SOA_Cluster

Table A-1 (Cont.) SOA Application Targets

Application	Targets
usermessagingdriver-apns	OSB_Cluster
usermessagingdriver-email	BAM_Cluster, OSB_Cluster, SOA_Cluster, MFT_Cluster
usermessagingdriver-extension	OSB_Cluster
usermessagingdriver-gcm	OSB_Cluster
usermessagingdriver-smpp	OSB_Cluster, MFT_Cluster
usermessagingdriver-xmpp	OSB_Cluster
usermessagingserver	BAM_Cluster, OSB_Cluster, SOA_Cluster, MFT_Cluster
worklistapp	SOA_Cluster
wsm-pm	WSM-PM_Cluster, MFT_Cluster
MFTCustomHostingApp	MFT_Cluster
MFTUI	MFT_Cluster
mftp-app	MFT_Cluster

Oracle SOA Enterprise Deployment Library Targets

This section provides a table that lists the Oracle SOA enterprise deployment library targets.

These are the default targets resulting from the EDG configuration. Other customizations (for example, extending the File Adapter in OSB) may require additional targeting.

Table A-2 SOA Library Targets

Library	Targets
adf.oracle.businesseditor#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.groovy#3.0.18@3.0.18	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.groovy.dateutil#3.0.18@3.0.18	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp#1.0@11.1.1.3.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster

Table A-2 (Cont.) SOA Library Targets

Library	Targets
adf.oracle.domain.webapp antlr-runtime#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.commons-lang3#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.httpclient#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.httpclient-cache#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.httpcore#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.httpcore-h2#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.httpmime#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.slf4j-api#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.apache.velocity#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.batik-bundle#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.graal-js#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.graal-js-scriptengine#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.graal-regex#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.graal-sdk#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.graal-truffle-api#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.guava#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster

Table A-2 (Cont.) SOA Library Targets

Library	Targets
adf.oracle.domain.webapp.icu4j#74@74.2	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.xml-apis-ext#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
adf.oracle.domain.webapp.xmlgraphics-commons#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
BamClientLibrary#12.2.1@12.2.1	BAM_Cluster
BamDatacontrol#12.2.1@12.2.1	BAM_Cluster
beam.em	AdminServer
emagentsdkimplpriv_jar#12.4@12.1.0.4.0	AdminServer
emagentsdkimpl_jar#12.4@12.1.0.4.0	AdminServer
emagentsdk_jar#12.4@12.1.0.4.0	AdminServer
emai.ess.fmwctrl.dep	AdminServer
emai.fmwctrl.dep	AdminServer
emas	AdminServer
emcore	AdminServer
emcoreclient_jar	AdminServer
emcorecommon_jar	AdminServer
emcoreconsole_jar	AdminServer
emcoreintsdk_jar#11.2.0.1.0@12.1.0.0.0	AdminServer
emcorepbs_jar	AdminServer
emcoresdkimpl_jar#11.2.0.1.0@12.1.0.0.0	AdminServer
emcoresdk_jar#11.2.0.1.0@12.1.0.0.0	AdminServer
emcore_jar	AdminServer
em_common#12.4@12.1.0.4.0	AdminServer
em_core_ppc_pojo_jar	AdminServer
em_error#12.4@12.1.0.4.0	AdminServer
em_sdkcore_ppc_public_pojo_jar	AdminServer
ess.em	AdminServer
JCAFrameworkImpl#12.1.2.0@12.1.2.0	AdminServer, OSB_Cluster
jsf#2.0@1.0.0.0_2-2-8	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
jsf1.2#1.2@1.2.9.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
jstl#1.2@1.2.0.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster

Table A-2 (Cont.) SOA Library Targets

Library	Targets
log4j-api_jar	AdminServer
log4j-bridge_jar	AdminServer
log4j-core_jar	AdminServer
odl.clickhistory#1.0@12.2.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
odl.clickhistory.webapp#1.0@12.2.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
ohw-rcf#5@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
ohw-uir#5@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.adapter.ext#12.1.2@12.1.2	AdminServer, SOA_Cluster
oracle.adf.dconfigbeans#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.adf.desktopintegration#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.adf.desktopintegration.model#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.adf.management#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.advancedanalytics.prediction#11.1.1@12.1.3	AdminServer, SOA_Cluster
oracle.bi.adf.model.slib#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.bi.adf.view.slib#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.bi.adf.webcenter.slib#1.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.bi.composer#11.1.1@0.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.bi.jbips#11.1.1@0.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.bpm.bac#11.1.1@12.1.3	AdminServer, SOA_Cluster
oracle.bpm.client#11.1.1@12.1.3	AdminServer, SOA_Cluster

Table A-2 (Cont.) SOA Library Targets

Library	Targets
oracle.bpm.composerlib#11.1.1@12.1.3	AdminServer, SOA_Cluster
oracle.bpm.management.webapp#12.1.3@12.1.3	AdminServer
oracle.bpm.processviewer#11.1.1@12.1.3	BAM_Cluster
oracle.bpm.projectlib#11.1.1@12.1.3	AdminServer, SOA_Cluster
oracle.bpm.runtime#11.1.1@12.1.3<	AdminServer, SOA_Cluster
oracle.bpm.webapp.common#11.1.1@12.1.3	AdminServer, SOA_Cluster
oracle.bpm.workspace#11.1.1@12.1.3	AdminServer, SOA_Cluster
oracle.cloud.adapter#12.1.2@12.1.2	AdminServer, SOA_Cluster
oracle.dconfig-infra#2.0@12.2.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.ess#12@14.1.2.0.0	AdminServer, ESS_Cluster, MFT_Cluster
oracle.ess.admin#12@14.1.2.0.0	AdminServer
oracle.ess.client#12@14.1.2.0.0	AdminServer, ESS_Cluster
oracle.ess.client.api#12@14.1.2.0.0	AdminServer, ESS_Cluster
oracle.ess.runtime#12@14.1.2.0.0	AdminServer, ESS_Cluster, MFT_Cluster
oracle.ess.thin.client#12@14.1.2.0.0	AdminServer, ESS_Cluster, OSB_Cluster, SOA_Cluster, MFT_Cluster
oracle.jrf.system.filter	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster
oracle.jsp.next#12.2.1@12.2.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster
oracle.mft#12.2.1.0@12.2.1.0	MFT_Cluster
oracle.mft.apache#12.2.1.0@12.2.1.0	MFT_Cluster
oracle.mft.bc#12.2.1.0@12.2.1.0	MFT_Cluster
oracle.mft.client#12.2.1.0@12.2.1.0	MFT_Cluster
oracle.pwdgen#2.0@12.2.1	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster
oracle.rules#11.1.1@12.1.3	AdminServer, SOA_Cluster
oracle.sdp.client#2.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.sdp.messaging#2.0@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.soa.apps#12.2.1@12.2.1	AdminServer, SOA_Cluster
oracle.soa.bpel#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.common.dvmtxref#12.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.common.functions#12.2.1@12.1.2	AdminServer, SOA_Cluster

Table A-2 (Cont.) SOA Library Targets

Library	Targets
oracle.soa.common.resequencer#12.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.common.sequencing#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.commonconsole.dependencies#12.1.2@12.1.2	AdminServer, SOA_Cluster, BAM_Cluster, MFT_Cluster
oracle.soa.commonconsole.webapp#12.1.2@12.1.2	AdminServer, SOA_Cluster, BAM_Cluster, MFT_Cluster
oracle.soa.composer.webapp#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.ess.dc#12@12.2.1.0.0	AdminServer, MFT_Cluster
oracle.soa.ext#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.management.webapp#12.1.2@12.1.2	AdminServer
oracle.soa.mediator#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.rules_dict_dc.webapp#11.1.1@11.1.1	AdminServer, SOA_Cluster
oracle.soa.rules_editor_dc.webapp#11.1.1@11.1.1	AdminServer, SOA_Cluster
oracle.soa.sb.em.adf.mgmt#1.0@12.1.2.0.0	AdminServer, OSB_Cluster
oracle.soa.webmapper#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.webmapper#12.1.3@12.1.3	AdminServer, OSB_Cluster
oracle.soa.workflow#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.workflow.wc#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.worklist#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.soa.worklist.webapp#11.1.1@11.1.1	AdminServer, SOA_Cluster
oracle.soa.xquery#11.1.1@12.1.2	AdminServer, SOA_Cluster
oracle.ucs.userprefs.webapp#2.0@14.1.2.0.0	BAM_Cluster, OSB_Cluster, SOA_Cluster, MFT_Cluster
oracle.webcenter.composer#2.0@14.1.2	AdminServer
oracle.webcenter.skin#2.0@14.1.2	AdminServer
oracle.wsm.console.core.view#1.0@14.1.2	AdminServer
oracle.wsm.idmrest.sharedlib#1.0@14.1.2	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
oracle.wsm.seedpolicies#2.0@14.1.2	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
orai18n-adf#11@11.1.1.1.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
osb.em	AdminServer
resource-mq-connection#12.1.3@12.1.3	AdminServer, OSB_Cluster
resource-mq-jms-connector#12.1.3@12.1.3	AdminServer, OSB_Cluster
soa.em	AdminServer
stage-logging#12.1.3@12.1.3	AdminServer, OSB_Cluster
stage-publish#12.1.3@12.1.3	AdminServer, OSB_Cluster

Table A-2 (Cont.) SOA Library Targets

Library	Targets
stage-routing#12.1.3@12.1.3	AdminServer, OSB_Cluster
stage-transform#12.1.3@12.1.3	AdminServer, OSB_Cluster
stage-utils#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-dsp#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-ejb#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-email#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-file#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-ftp#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-jca#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-jejb#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-mq#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-pollersdk#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-sftp#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-soa#12.1.3@12.1.3	AdminServer, OSB_Cluster
transport-tuxedo#12.1.3@12.1.3	AdminServer, OSB_Cluster
UIX#11@14.1.2.0.0	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster

Oracle SOA Enterprise Deployment Startup Class Targets

This section provides a table that lists the Oracle SOA enterprise deployment Startup Class targets.

These are the default targets resulting from the EDG configuration. Other customizations may require additional targeting.

Table A-3 SOA Startup Class Targets

Class	Targets
AWT Application Context Startup Class	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
DMS-Startup	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
JRF Startup Class	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
ODL-Startup	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
OSB JCA Transport Post-Activation Startup Class	OSB_Cluster, AdminServer

Table A-3 (Cont.) SOA Startup Class Targets

Class	Targets
SOAStartupClass	SOA_Cluster
Web Services Startup Class	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
WSM Startup Class	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster

Oracle SOA Enterprise Deployment Shutdown Class Targets

This section provides a table that lists the Oracle SOA enterprise deployment Shutdown Class targets.

These are the default targets resulting from the EDG configuration. Other customizations may require additional targeting.

Table A-4 SOA Shutdown Class Targets

Class	Targets
DMSShutdown	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster

Oracle SOA Enterprise Deployment JMS System Resource Targets

This section provides a table that lists the Oracle SOA enterprise deployment JMS System Resource targets.

These are the default targets resulting from the EDG configuration. Other customizations may require additional targeting.

Table A-5 SOA JMS System Resource Targets

JMS Resource	Targets
BamAlertEngineJmsSystemModule	BAM_Cluster
BamCQServiceJmsSystemModule	BAM_Cluster
BAMJMSSystemResource	BAM_Cluster
BamPersistenceJmsSystemModule	BAM_Cluster
BamPersistenceJmsSystemModule	BAM_Cluster
BamReportCacheJmsSystemModule	BAM_Cluster
BPMJMSSModule	SOA_Cluster
jmsResources	OSB_Cluster

Table A-5 (Cont.) SOA JMS System Resource Targets

JMS Resource	Targets
OSBAQJMSServer	
SOAJMSModule	SOA_Cluster
UMSJMSSystemResource	BAM_Cluster, OSB_Cluster, SOA_Cluster
UMSAQJMSSystemResource	MFT_Cluster
MFTJMSModule	MFT_Cluster

Oracle SOA Enterprise Deployment JDBC System Resource Targets

This section provides a table that lists the Oracle SOA enterprise deployment JDBC System Resource targets.

These are the default targets resulting from the EDG configuration. Other customizations may require additional targeting.

Table A-6 SOA JDBC System Resource Targets

JDBC Resource	Targets
BamDataSource	BAM_Cluster
BamJobSchedDataSource	BAM_Cluster
BamLeasingDataSource	BAM_Cluster
BamNonJTADDataSource	BAM_Cluster
EDNDataSource	SOA_Cluster
EDNLocalTxDataSource	SOA_Cluster
EssDS	ESS_Cluster, MFT_Cluster
EssInternalDS	ESS_Cluster, MFT_Cluster
EssXADS	ESS_Cluster, MFT_Cluster
LocalSvcTblDataSource	AdminServer, ESS_Cluster
mds-bam	AdminServer, BAM_Cluster
mds-ESS_MDS_DS	ESS_Cluster, MFT_Cluster
mds-owsm	AdminServer, WSM-PM_Cluster, MFT_Cluster
mds-soa	AdminServer, SOA_Cluster
opss-audit-DBDS	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
opss-audit-viewDS	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster
opss-data-source	AdminServer, BAM_Cluster, ESS_Cluster, OSB_Cluster, SOA_Cluster, WSM-PM_Cluster, MFT_Cluster

Table A-6 (Cont.) SOA JDBC System Resource Targets

JDBC Resource	Targets
OraSDPMDDataSource	BAM_Cluster, OSB_Cluster, SOA_Cluster, MFT_Cluster
SOADDataSource	AdminServer, OSB_Cluster, SOA_Cluster
SOALocalTxDataSource	SOA_Cluster
wlsbjmsrpDataSource	AdminServer, OSB_Cluster
WLSRuntimeSchemaDataSource	AdminServer, OSB_Cluster, SOA_Cluster, BAM_Cluster, MFT_Cluster
WLSSchemaDataSource	AdminServer, OSB_Cluster, SOA_Cluster, BAM_Cluster, MFT_Cluster
MFTDataSource	MFT_Cluster
MFTLocalTxDataSource	MFT_Cluster
mds-mft	AdminServer, MFT_Cluster