Oracle® Communications Order and Service Management

Installation Guide

Release 8.0
G37994-01
October 2025

Oracle Communications Order and Service Management Installation Guide, Release 8.0

G37994-01

Contents

8 Troubleshooting OSM Installation Problems

9 Verifying the OSM Installation

12 Updating Order-to-Activate Cartridges

C OSM Development System Guidelines and Best Practices

D OSM Installer Properties

E Installing OSM on Engineered Systems

F Installing OSM on Oracle Cloud Infrastructure

About This Content

This document describes how to install Oracle Communications Order and Service Management (OSM).

Audience

This document is intended for system administrators, system integrators, database administrators, and other individuals who are responsible for installing OSM and ensuring that the software is operating in the manner required for your business. This guide assumes that users have a good working knowledge of the operating systems they will be using, the Oracle Database, Oracle WebLogic Server, and Java J2EE software.

Documentation Accessibility

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

Access to Oracle Support

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

Conventions

The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

OSM Production Installation Overview

This chapter provides an overview of the Oracle Communications Order and Service Management (OSM) production installation process.

Overview of OSM Installed Components

During the installation process, you install and configure the following components:

- Oracle Grid Infrastructure
- Oracle Database
- Oracle Fusion Middleware Infrastructure that includes Oracle WebLogic Server, Oracle Coherence, and Oracle Application Development Framework (ADF)
- Ant
- OSM server software and OSM clients

Overview of the OSM Production Installation Procedure

The OSM production installation should be performed only by qualified personnel. You must be familiar with the operating system on which OSM is to be installed and with WebLogic Server. The installation and configuration of Oracle Database should be performed by an experienced database administrator.

You set up OSM in your production environment by performing the following tasks:

1. Plan your OSM production installation. See "Planning Your OSM Production Installation ".
2. Ensure your system meets the minimum system requirements. See "OSM System Requirements" for system requirements and configuration information you need before installing OSM.
3. Perform pre-installation tasks such as:
    a. Install and configure the Oracle RAC Database Server. See "Installing and Configuring the Oracle RAC Database".
    b. Install and configure Oracle WebLogic Server Cluster. See "Installing and Configuring the WebLogic Server Cluster".
4. Install OSM. See "Installing OSM".
5. Perform post-installation tasks. See "Performing OSM Post-Installation Tasks ".
6. Verify the installation. See "Verifying the OSM Installation".
7. Test and Tune the OSM Installation. See "OSM Pre-Production Testing and Tuning".

This document is intended to help you install an OSM production environment. However, you can use OSM with the default installation settings for product evaluation, development, demonstration, or process modeling. For more information, see OSM Development System Guidelines and Best Practices.

Installing OSM with Dual Stack Enabled (IPv4/IPv6) Linux System

OSM is supported on an IPv6 environment. You can configure OSM to run using an IPv6 address on a server with dual IPv4/IPv6 addresses.

The OSM installation process is similar to an IPv4-based installation, except that IPv6 addresses must be entered using the standard square bracket [] notation.

IPv6 addresses must be enclosed within square brackets “[]" during the following procedures:

- Creation of Weblogic cluster, by providing an IPv6 address
- Installation of OSM with IPv6 address
- Launch of OSM Task web client
- Cartridge deployment through CMT and Design Studio

Ensuring a Successful OSM Installation

As you install each component (for example, Oracle Database and WebLogic Server), verify that the component installed successfully before continuing the installation.

Pay close attention to the system requirements. Before you begin installing software, ensure that your system has the required base software. In addition, ensure that you know all of the required configuration values, such as host names and port numbers.

As you create new configuration values, write them down. In some cases, you might need to re-enter configuration values later in the procedure. Also, some values will need to be communicated to all users of OSM.

Directory Placeholders Used in This Guide

The following placeholders are used in this guide:

Table 1-1 Placeholders Used in This Guide

Placeholder	Directory Description
OSM_home	The directory in which the OSM software is installed.
Middleware_home	The location where Oracle Fusion Middleware components are installed. This directory contains the base directory for WebLogic Server and the **oracle_common** directory, among other files and directories.
WebLogic_home	The base directory for the WebLogic Server core files. It is located in the *Middleware_home* directory; for example, *Middleware_home***/wlserver**.
domain_home	The directory that contains the configuration for the domain into which OSM is installed. The default is *Middleware_home***/user_projects/domains/***domain_name* (where *domain_name* is the name of the OSM domain), but it is frequently set to some other directory at installation.

2

OSM System Requirements

This chapter describes the Oracle Communications Order and Service Management (OSM) system requirements. It also describes information you need during the installation procedure.

Software Requirements

For details about the software required to support the OSM components for traditional deployment, see "OSM Traditional Deployment Software Compatibility" in *OSM Compatibility Matrix*.

Information Requirements

You must supply configuration-related information when you install OSM. Some of this information is determined during the installation. However, some information should already have been determined before the OSM installation is begun. Gather this information in advance.

> **Note**
>
> You can obtain most of the required configuration information from your Oracle database administrator (DBA) and your system administrator.

Oracle Database Information

The information in Table 2-1 is needed to connect to an Oracle database instance.

Table 2-1 Oracle Database Connection Information

Information Type	Description
Database Instance Host	The IP address or DNS name of the database instance on which the OSM schema will be installed. If you plan to use an Oracle RAC database, you need the Single Client Access Name (SCAN) for this field.
Database Instance Port	The port on which the database instance to which the OSM schema will be installed is listening.
Database Instance SID	The SID of the database instance on which the OSM schema will be installed.
Database Service Name	The name of the service used for OSM client connections.

If the OSM database users have been created prior to the OSM installation, you must gather the information in Table 2-2 before installing OSM. If they have not been created, you can determine the following information during the installation.

Table 2-2 Oracle Database Credential Information

Information Type	Description
Core Schema User name	The OSM core schema username.
Core Schema Password	The OSM core schema username password.
Rule Engine Schema User name	The OSM rule engine schema username.
Rule Engine Schema Password	The OSM rule engine schema username password.
Reporting Schema User name	The OSM Reporting Interface schema username.
Reporting Schema Password	The OSM Reporting Interface schema username password.
Database Admin User name	The user name of the Oracle DBA.
Database Admin Password	The password of the Oracle DBA.

Table 2-3 shows the information used to indicate the tablespaces being used for the installation.

Table 2-3 Oracle Database Tablespace Information

Information Type	Description
Default Tablespace	The tablespace to use if your data will all be on the same permanent tablespace.
Temporary Tablespace	If your database instance has only one temporary tablespace, the installer will find its name automatically. If there is more than one temporary tablespace in the database instance, you must know which one OSM is supposed to use.
Model Data Tablespace	The name of the tablespace to use for model data, if different from the default tablespace.
Model Index Tablespace	The name of the tablespace to use for the indexes for the model data, if different from the default tablespace.
Order Data Tablespace	The name of the tablespace to use for order data, if different from the default tablespace.
Order Index Tablespace	The name of the tablespace to use for the indexes for the order data, if different from the default tablespace.

WebLogic Server Information

Table 2-4 shows the WebLogic Server information needed to connect to a WebLogic server.

Table 2-4 WebLogic Server Information

Data	Description
WebLogic Host Name	The name or IP address of the computer where WebLogic Server is installed.
WebLogic Port Number	The port number of the WebLogic server.
WebLogic SSL Port Number	The SSL port number of the WebLogic server.
WebLogic Admin User	The user name of the WebLogic system administrator.

Table 2-4 (Cont.) WebLogic Server Information

Data	Description
WebLogic Admin Password	The password of the WebLogic system administrator.

OSM Server Information

Table 2-5 shows the information related to the OSM server installation. The following information is used to specify the SMTP mail server and the administrative e-mail addresses to deliver notifications via e-mail.

Table 2-5 Email Notification

Data	Description
Notification Email server	DNS name or IP address of your e-mail server.
Notification Email server Port	Port that the e-mail server is listening.
Administrator Email Address	OSM Administrator's e-mail address.

3

Planning Your OSM Production Installation

This chapter describes how to plan an Oracle Communications Order and Service Management (OSM) production installation.

Overview of Planning Your OSM Production Installation

This section provides an overview of the planning process that you must perform to identify and acquire the hardware, software, and networking equipment and configurations required to run OSM in a highly available production environment.

When you plan your installation, you must consider such options as the type of OSM system you need, the types of orders you must process, the hardware you need, the amount of memory, CPU, and I/O required, the networking equipment and configuration needed, the database, WebLogic Server, and operating system configuration requirements.

When planning for your production installation, you should determine whether there is a more recent patch of OSM available that you want to use for your production system. For more information, see "Checking for a Current OSM Patch Before Going into Production."

Types of Systems

Planning an OSM installation depends on the purpose of the OSM system. You can define the following OSM systems:

- **Development Systems**: The purpose of a development system is to develop, deploy, test, and demonstrate new OSM solution functionality. For more information about development systems, see "OSM Development System Guidelines and Best Practices."
- **Production Systems**: The purpose of a production system is to process orders in an overall OSM solution. Production systems must be highly available, scalable, and secure. Before you go live with a production system, you must simulate the production system environment and expected order volume as closely as possible. You can use a pre-production environments to generate performance data, to test system tuning procedures, and to provide a staging environment before moving to a production system.

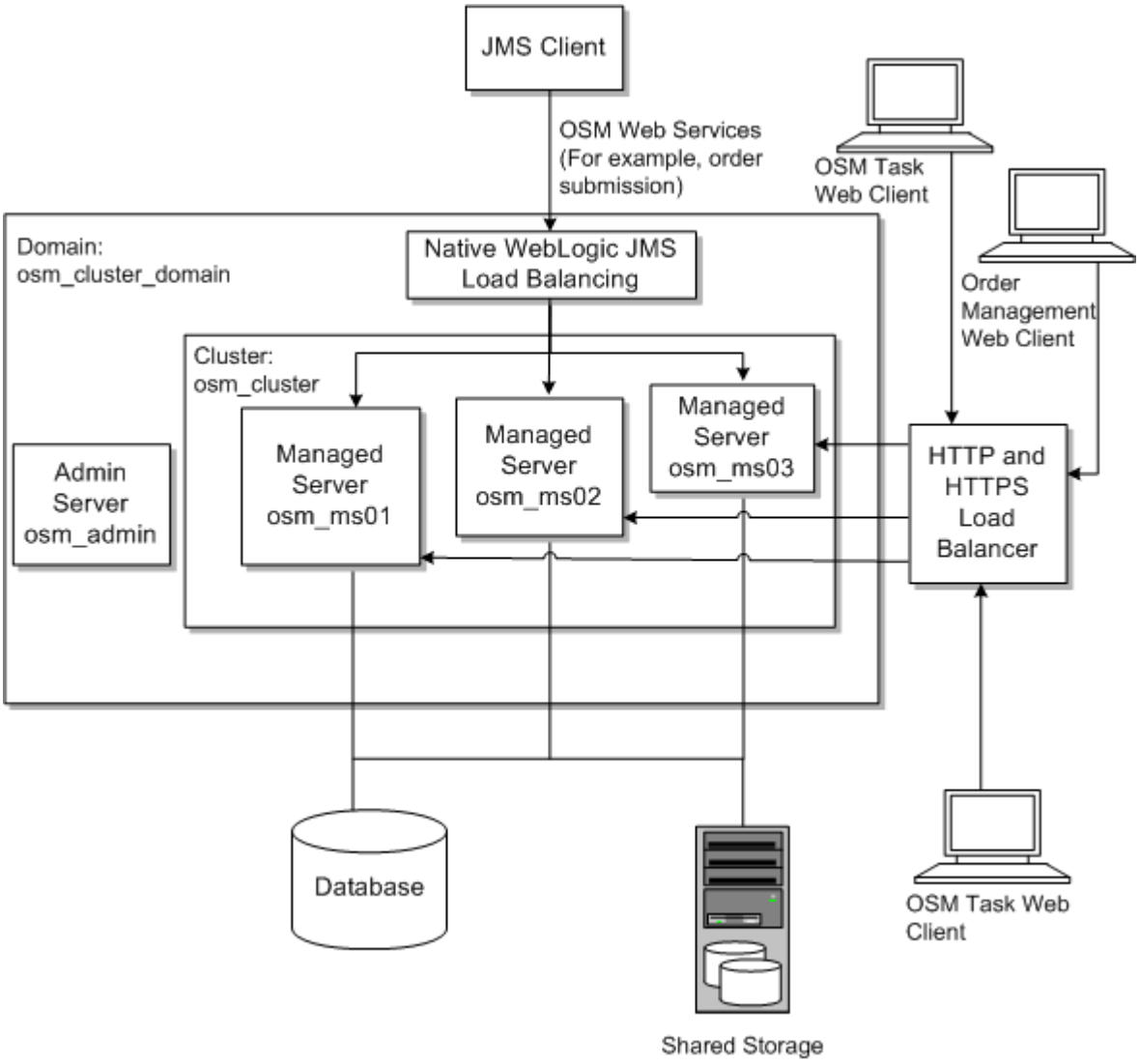
High Availability Architecture

In a highly-available OSM deployment, redundancy is built in at each layer to ensure there is no single point of failure. A OSM system that is deployed into a high-availability architecture consists of the following:

- The application server layer, which hosts the WebLogic Server cluster that manages the OSM application.
- The database server layer, which hosts a highly-available Oracle RAC database instances.
- A shared storage system that the database servers use to access the database files.
- A shared storage system for the application servers for whole server migration in case of a server failure.
- An HTTP load balancer in fault-tolerant mode.

Figure 3-1 shows an example of a highly-available OSM production system topology. This system is deployed across multiple physical servers at the application server layer.

Figure 3-1 OSM High-Availability Test or Production System Topology



The system includes a WebLogic Server cluster with four managed servers, an administration server, a JMS client sending orders to OSM, and an HTTP load balancer that load balances HTTP and HTTPS messages to OSM from various OSM web clients. Each physical server can host one or more managed servers. The managed servers form the WebLogic server cluster that runs OSM. At the database server layer, OSM supports a partitioned active-active deployment of two or more Oracle Real Application Clusters (Oracle RAC) instances with shared storage.

For increased availability for the WebLogic Server, Oracle recommends that you configure managed servers in the cluster with whole server migration. Whole server migration enables a managed server that unexpectedly terminates and cannot be restarted to migrate and start up on a different machine. The standby machine can be empty or host an existing OSM managed server. If the machine hosts an existing managed server, then the machine must have enough capacity to run a second OSM managed server. You must acquire and configure shared

storage to ensure the persistence of managed server data when a managed server migrates to another machine.

Initial Sizing Based on Order Complexity and Performance Needs

The size of an OSM production system depends on the overall complexity of the solution. Complexity can be determined using criteria such as:

- The average number of orders per day
- The order creation rate during peak hour
- The number of days the order must be retained
- The number of order line items per order
- The number of order components per order
- The number of tasks per order
- The number of data elements and their complexity per order
- The number of tasks per second (throughput)
- Expected order lifetime (from creation to completion in seconds)
- Number of manual users

The most common measure of OSM performance is order throughput. OSM must fulfill orders at the rate that is determined by business need. OSM throughput is measured in task transitions per second (TPS).

Although the TPS metric varies for each deployment, it is useful for you to consider the following approximate guidelines and adjust them as your circumstances require:

- **Simple orders**, which typically complete less than 10 tasks per order.
- **Moderate orders**, which complete approximately 25 tasks per order.
- **Complex orders**, which complete approximately 100 tasks per order.
- **Very complex orders**, which complete approximately 1000 tasks per order.

Given this criteria, you can create an initial estimate of the hardware requirements for an OSM production installation. See "Planning the Physical Architecture" for more information about estimating hardware requirements.

> **Note**
>
> The solution architecture impacts the size of orders, the number of tasks, and so on. This enables you to do the initial sizing, but you must still confirm this sizing with actual performance testing. For more information, see "OSM Pre-Production Testing and Tuning."

Planning the Solution Architecture, System Deployment, and Maintenance

A solution architecture, which refers to the structure, interaction, and abstraction of software applications, needs to be devised. It represents how the various components, including OSM, interact. It also showcases issues such as:

- What role does each component or product play in the solution?

- Who can access and use this component?
- How is it secured?

For more information, see *OSM Modeling Guide*.

In addition to the hardware and software requirements, you must also plan system deployment and ongoing maintenance considerations such as:

- How will the OSM in the central order management (COM), service order management (SOM), and technical order management (TOM) roles map to the physical architecture? Will there be multiple instances of OSM?

  OSM in the COM role is typically deployed in a separate WebLogic Server cluster and OSM in the SOM and TOM roles are typically in the same WebLogic Server clusters. You must do performance tests for each OSM instance in your network. Oracle recommends that OSM is deployed on dedicated machines. Sharing machines makes troubleshooting much more difficult.
- What is the backup and restore strategy?
- How will application performance monitoring and operational maintenance activities, such as log file clearing, be handled?
- How will business continuity be maintained in the event of an application failure?
- What are the data retention requirements and the data purge strategy?
- How will errors and failures in other interfacing applications be handled?
- How will the production environment be deployed? Is there a requirement for automated deployment?

Planning the Physical Architecture

The following sections describe hardware sizing guidelines for OSM using RODOD, RSDOD, and very simple non-RODOD or RSDOD solution examples on Oracle Linux. These sections provide guidelines for estimating the hardware required to achieve similar daily order volumes with your own OSM solutions. It also includes general sizing guidelines applicable to any solution type.

These guidelines are intended to assist in estimating the total OSM system requirements in very early stages of implementation, before OSM is installed. These guidelines do **not** contain express or implied warranties of any kind. After you install OSM and build a solution, you must do performance tests to validate whether the hardware selected is enough for production order volumes. For more information, see "OSM Pre-Production Testing and Tuning."

The hardware recommendations below cover only the OSM part of the solution. For all recommendations a day is considered to be 10 hours. The database storage service time is expected to be less than 5 milliseconds. Each managed server in the cluster has a total heap of just under 32 GB. The values in the tables are approximate.

Generally, the more complex your orders are, the fewer orders your OSM system can process per day, per managed server, and per database instance. To increase the number of orders you can process per day, you can either:

- Simplify your orders, or
- Configure additional managed servers, database instances, and hardware.

OSM COM Hardware Sizing Guidelines for RODOD Orders

You can use two models for RODOD COM sizing: simple and complex. They are different in number of sales lines and components, and, as a result, in the number of tasks and number of orders that can be completed per day.

Sizing Guidelines for Simple RODOD COM Orders

Table 3-1 shows sizing guideline assumptions for simple RODOD COM orders. One order contains:

- 20 automated tasks
- five components
- five sales lines

Table 3-1 Hardware Sizing Guidelines for Simple RODOD COM Orders

Deployment Size	Server	Small	Medium	Large
Orders/day	Not Applicable	<= 250,000	250,000 <= 1,000,000	1,000,000 <= 2,000,000
Reference server model	Application	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2
Number of servers (HA configuration in parentheses)	Application	1 (2)	1 (2)	2 (3)
CPUs	Application	1 Intel(R) Xeon(R) Gold 5222 3.8 GHz (4 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.4 GHz (24 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.4 GHz (24 cores)
RAM (DDR4, GB)	Application	64	192	192
Internal disk space (GB)	Application	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)
Number of WebLogic Server managed servers (HA configuration in parentheses)	Application	2 (4)	4 (8)	8 (12)
Shared storage IOPS (for failover purposes), total for all nodes	Application	5000	17500	35000
Reference server model	Database	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2
Number of servers (HA configuration in parentheses)	Database	1 (2)	1 (2)	2 (3)
CPUs	Database	1 Intel(R) Xeon(R) Gold 5222 3.8 GHz (4 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.4 GHz (24 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.4 GHz (24 cores)
RAM (DDR4, GB)	Database	32	128	128
Database storage IOPS, total for all nodes	Database	10000	35000	70000

Sizing Guidelines for Complex RODOD COM Orders

Table 3-2 shows sizing guideline assumptions for complex RODOD COM orders. One order contains:

- 40 automated tasks
- 15 components
- 20 sales lines

Table 3-2 Hardware Sizing Guidelines for Complex RODOD COM Orders

Deployment Size	Server	Small	Medium	Large
Orders/day	Not Applicable	<= 50,000	50,000 <= 200,000	200,000 <= 400,000
Reference server model	Application	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2
Number of servers (HA configuration in parentheses)	Application	1 (2)	1 (2)	2 (3)
CPUs	Application	2 Intel(R) Xeon(R) Gold 5222 3.80 GHz (4 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)
RAM (DDR4, GB)	Application	128	128	128
Internal disk space (GB)	Application	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)
Number of WebLogic Server managed servers (HA configuration in parentheses)	Application	2 (4)	4 (8)	8 (12)
Shared storage IOPS (for failover purposes), total for all nodes	Application	5000	17500	35000
Reference server model	Database	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2
Number of servers (HA configuration in parentheses)	Database	1 (2)	1 (2)	2 (3)
CPUs	Database	2 Intel(R) Xeon(R) Gold 5222 3.80 GHz (4 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)
RAM (DDR4, GB)	Database	64	128	128
Database storage IOPS, total for all nodes	Database	10000	35000	70000

OSM SOM Hardware Sizing Guidelines for RSDOD Orders

Table 3-3 shows sizing guideline assumptions for RSDOD SOM orders. One order contains:

- 20 automated tasks
- Five components
- 10 sales lines

Table 3-3 Hardware Sizing Guidelines for RSDOD SOM Orders

Deployment Size	Server	Small	Medium	Large
Orders/day	Not Applicable	<= 250,000	250,000 <= 1,000,000	1,000,000 <= 2,000,000
Reference server model	Application	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2
Number of servers (HA configuration in parentheses)	Application	1 (2)	1 (2)	2 (3)
CPUs	Application	1 Intel(R) Xeon(R) Gold 5222 3.80 GHz (4 cores)	1 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)	1 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)
RAM (DDR4, GB)	Application	64	192	192
Internal disk space (GB)	Application	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)
Number of WebLogic Server managed servers (HA configuration in parentheses)	Application	2 (4)	4 (8)	8 (12)
Shared storage IOPS (for failover purposes), total for all nodes	Application	5000	17500	35000
Reference server model	Database	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2
Number of servers (HA configuration in parentheses)	Database	1 (2)	1 (2)	2 (3)
CPUs	Database	2 Intel(R) Xeon(R) Gold 5222 3.80 GHz (4 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)	2 Intel(R) Xeon(R) Platinum 8260 2.40 GHz (24 cores)
RAM (DDR4, GB)	Database	32	128	128
Database storage IOPS, total for all nodes	Database	10000	35000	70000

Simple Order Hardware Sizing Guidelines (Neither RODOD nor RSDOD)

Table 3-4 shows sizing guideline assumptions for simple (neither RODOD nor RSDOD) orders. One order contains:

- Five automated tasks
- Five components
- 10 sales lines

Table 3-4 Hardware Sizing Guidelines for Simple Orders

Deployment Size	Server	Small	Medium	Large
Orders/day	Not Applicable	<= 2,000,000	2,000,000 <= 5,000,000	5,000,000 <= 10,000,000
Reference server model	Application	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2

Table 3-4 (Cont.) Hardware Sizing Guidelines for Simple Orders

Deployment Size	Server	Small	Medium	Large
Number of servers (HA configuration in parentheses)	Application	1 (2)	1 (2)	2 (3)
CPUs	Application	1 Intel(R) Xeon(R) Gold 5222, 3.80 GHz (4 cores)	1 Intel(R) Xeon(R) Platinum 8260, 2.40 GHz (24 cores)	1 Intel(R) Xeon(R) Platinum 8260, 2.40 GHz (24 cores)
RAM (DDR4, GB)	Application	64	192	192
Internal disk space (GB)	Application	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)	2x600 SAS-3 HDD (RAID1)
Shared storage IOPS (for failover purposes), total for all nodes	Application	500	1000	2000
Number of WebLogic Server managed servers (HA configuration in parentheses)	Application	2 (4)	4 (8)	8 (12)
Reference server model	Database	Oracle Server X8-2	Oracle Server X8-2	Oracle Server X8-2
Number of servers (HA configuration in parentheses)	Database	1 (2)	1 (2)	2 (3)
CPUs	Database	1 Intel(R) Xeon(R) Gold 5222, 3.80 GHz (4 cores)	2 Intel(R) Xeon(R) Gold 5222, 3.80 GHz (4 cores)	1 Intel(R) Xeon(R) Platinum 8260, 2.40 GHz (24 cores)
RAM (DDR4, GB)	Database	32	128	128
Database storage IOPS, total for all nodes	Database	3000	6500	13000
Number of Oracle RAC nodes (HA configuration in parentheses)	Database	1 (2)	1 (2)	2 (3)

General Hardware Sizing and Configuration Recommendations

The following sections provide general hardware sizing and configuration recommendations.

OSM Installer and Application Server System Sizing

After performing the prerequisite tasks such as installing Fusion Middleware and Java, ensure you have a minimum of 10 GB of available disk space for installing and deploying all OSM required packages, creating a domain, and deploying the OSM application.

Application Server Hardware Sizing

Your cluster should be equally distributed across the number of Oracle RAC database instances that OSM uses. For example, if your order volume requirements mandates that you need three managed servers and two Oracle RAC database instances, you must round up the number of managed servers to four so that each database instance has an equal number of

managed servers. Each managed server should have a total heap of just under 32 GB of memory. Oracle recommends that you use the managed server startup parameters and memory configuration specified in "Configuring Managed Server Startup Parameters."

Running Multiple WebLogic Servers on the Same System

You can run multiple WebLogic servers in a cluster on the same system so as to maximize the use of available resources while keeping the heap size of the associated JVM instances to a reasonable level. Ensure that you limit the number of JVM instances based on the number of available processors.

Shared Storage for the WebLogic Server

In an OSM high-availability system architecture, if transaction logs and JMS messages are not persisted in JDBC stores, the WebLogic Server requires high-performance disk storage to support whole server migration. Whole server migration enables a managed server that fails on one system to migrate and startup on another system. Various files must be installed on shared storage across WebLogic managed server instances such as persistent stores.

See "Understanding Whole Server Migration for High Availability" for information about the WebLogic Server files that must be on shared storage to support the whole server migration functionality. All other files can be on the local file system where the managed servers or administration server are running.

> **Note**
>
> Other shared storage file configurations are possible depending on your business requirements. Check Fusion Middleware Documentation for information on best practices for Weblogic Server on shared storage.

Use RAID 1+0 (normal redundancy) backed shared storage for installing WebLogic Server, creating the OSM domain, and deploying applications and server log files. Ensure that you have 5 GB for JMS persistent file stores. This requirement might be higher, depending on the design and the order volume.

Database Hardware Sizing

For the database, plan to have at least 500 GB of free disk space for the OSM schema. The OSM schema size is likely to be higher, depending on the design and the order volume, and you must plan for this during hardware sizing. The size of the OSM schema depends many factors such as the number orders you process per day, the duration you must retain the orders, and so on.

Shared Storage for the Database

In an OSM high-availability system architecture, the Oracle database requires high-performance shared storage to support database failover operations. In an Oracle RAC configuration, all data files, control files, and parameter files are shared for use by all Oracle RAC instances.

A high-availability storage solution that uses one of the following architectures is recommended:

- Direct Attached Storage (DAS), such as a dual ported disk array or a Storage Area Network (SAN).
- Network Attached Storage (NAS).

In terms of I/O characteristics, OSM performs a large amount of database writes compared to database reads. OSM is highly sensitive to writes performance of the Oracle Database. Ensure that the write response times of the storage do not exceed 5ms under the load.

You should also consider backup and restore hardware and software that supports mirror split and re-mirroring such as the Oracle ZFS Storage Appliance. Oracle recommends such mirroring software and hardware because the backup and restore functionality is rapid and can be done online. This functionality is especially important when performing upgrades or when purging database partitions and can reduce the length of maintenance windows or the time it takes to recover from errors. For more information about backing up and restoring OSM files and data, see *OSM System Administrator's Guide*.

RAID Recommendations for the Database

Redundant Array of Independent Disks (RAID) disk storage technology increases disk storage functions and reliability through redundancy. The use of RAID with redundant controllers is recommended to ensure there is no single point of failure and to provide better performance on the storage layer.

The database is sensitive to read/write performance of the redo logs and should be on a RAID 1, RAID 1+0, or no RAID at all because logs are accessed sequentially and performance is enhanced by having the disk drive head near the last write location.

See the *I/O Tuning with Different RAID Configurations* (Doc ID 30286.1) knowledge article on the Oracle support website for additional information:

https://support.oracle.com

Understanding Order Affinity

The following section provides information about load balancing.

About Order Affinity and Ownership in an OSM WebLogic Cluster

When an OSM managed server receives a new order, OSM assigns a unique order ID to the order. OSM associates the order ID to the receiving managed server instance name within the cluster. Throughout the order fulfillment life cycle, OSM processes this order only with the associated managed server. This OSM principle is called *order affinity* and ensures order data integrity and performance by preventing multiple managed server instances from processing the same order. The server instance that has control of an order owns the order. OSM routes all requests relating to the order to the owner instance.

Order ownership is transferable. OSM can transfer an order to another managed server in the following scenarios:

- If an order becomes a high-activity order, OSM can redistribute the order from the receiving managed server to another less-active managed server to better balance the load between each server in the cluster (see "Distribution of High-Activity Orders" for more information).
- If an incoming order is a revision order that arrives on a managed server different from the one processing the base order, OSM transfers order ownership so that the same managed server owns both the base order and the incoming order.

- If the incoming order has a dependency on an order owned by a server instance other than the one on which it was received. For example, a follow-on order that has a dependency on another order would be routed to the server where the previous order was processed.
- Before redistribution of an order to a new or different server instance, that server instance notifies other server instances to complete pending operations on the orders to be redistributed and delete them from their order cache.

> **Note**
>
> The reassignment of orders can temporarily impact Oracle RAC database performance when order ownership changes as the OSM WebLogic cluster resizes.

- If a managed server is added or removed from a cluster, OSM notifies all server instances about topology changes to the cluster and re-runs the distribution algorithms that determine which server instance owns an order. Order ownership either remains with the previous owner or with a different owner.

> **Note**
>
> The user will be logged out of the web client and will have to log back in, if you have a WebLogic Server cluster, and the following conditions apply:
>
> - a user is viewing an order in the Order Management web client or Task web client
> - that order is hosted on a managed server that fails or is shut down

About Load Balancing for OSM and Order Affinity

Load balancing helps maximize server resource use and order throughput. It enables OSM to minimize server response time and processing delays that can occur if some servers are overloaded while others remain unused. Load balancing helps support rolling downtimes of servers for maintenance tasks or upgrade procedures without impacting clients during non-peak times.

For OSM, two types of incoming messages are important for load balancing:

- Load balancing for JMS over T3 or T3S. Inbound JMS messages to OSM can include:
  - OSM Web Service requests, such as Create Order requests from a CRM that initiates an OSM order
  - JMS messages responding to an OSM automation, such as a response to an automation plug-in JMS request messages to external fulfillment systems
- Load balancing for HTTP and HTTPS. Inbound HTTP and HTTPS messages to OSM can include:
  - OSM Web Service requests transmitted over HTTP and HTTPS, such as CreateOrder requests from a CRM that initiates an OSM order
  - OSM web client interactions, including the Task web client and Order Management web client
  - XML API requests from external system

> **Note**
>
> OSM automations often use the XML API function calls while processing orders within a local server instance. However, OSM typically uses the XML API locally on the same server instance, because the XML API is often used to manipulate the same order owned by the local instance.

For JMS messages, OSM uses the Oracle WebLogic Server JMS distributed destinations for load balancing. See "JMS Distributed Destinations" for more information. You do not need to load balance JMS messages using an external load balancer.

For HTTP and HTTPS messages, Oracle recommends using a software or hardware load balancer outside of the OSM WebLogic cluster.

Load balancing for OSM Web Service requests is important because the OSM order affinity functionality requires that the orders are distributed appropriately among each managed server within the cluster. A managed servers that receive an order becomes the owner of the order. See "About JMS Load Balancing Schema Options" for more information about JMS load-balancing options and see "About HTTP and HTTPS Load Balancing and Session ID Configuration" for more information about HTTP and HTTPS load balancing options.

About the Performance Differences Between JMS and HTTP or HTTPS

In some order affinity scenarios, OSM must forward the requests from a receiving managed server to the owner managed server, such as when a CRM system sends a revision order and OSM receives the order on a managed server that is not the owner of the base order. This process is different depending on whether the message is delivered over JMS or over HTTP or HTTPS.

If the CRM sends the revision order over JMS, OSM re-directs the request to the owner instance. The managed server that originally received the order no longer participates in any way in the processing of the order.

If the CRM sends the revision order over HTTP or HTTPS, OSM forwards the request to the owner managed server over the internal JMS messaging queues. However, the receiving managed server must continually maintain a socket connection to the HTTP client or load balancer that sent the revision order even though another managed server is responsible for processing both the revision order and the base order. The socket connection on the receiving server must remain open until a response is generated because HTTP messages are synchronous. This restriction adds a performance overhead when sending orders over HTTP or HTTPS and increases with the size of the WebLogic Server cluster, because the probability of a message, like a revision order, arriving at the server that owns a particular order decreases as the ownership of orders is spread across more servers.

Given the above limitation, as well as the advantage of the transactional reliability of JMS message processing, Oracle recommends using the OSM Web Services over JMS messages for external client communication. Use HTTP and HTTPS messages for the OSM Order Management web client and the OSM Task web client because human interaction with these clients are synchronous by nature.

About Order Affinity and Ownership in an Oracle RAC Database

WebLogic multi data sources support XA affinity for global transactions, which ensures that all the database operations for a global transaction performed on an Oracle RAC cluster are directed to the same Oracle RAC instance. However, XA affinity cannot span different global

transactions on the same data, which is a key performance requirement for OSM. The objective is to minimize the adverse impact on performance caused by database buffer cache transfers between Oracle RAC instances. Therefore, OSM supports order affinity, which means that database operations for different global transactions on the same order are normally directed to the same Oracle RAC instance. Overall, Oracle RAC instances process database operations simultaneously, but each instance operates on a subset of orders mutually exclusive to each other.

OSM order affinity works in the following way:

- Each OSM server interacts with Oracle RAC through a WebLogic Server multi data source configured for failover with two data sources (one for each Oracle RAC instance). This setup is used for both active-passive and active-active topologies. Under normal conditions each OSM server always interacts with a single Oracle RAC instance. In an active-active topology, load balancing is achieved by reversing the order of the data sources in the multi data source for half of the OSM servers.
- If the Oracle RAC database is configured with server-side load balancing (the Oracle RAC instances register with a remote listener process), server-side load balancing must be overridden as discussed in "Remote Listener Considerations."
- Under normal conditions, the ownership and processing of each order is pinned to a single OSM server in a cluster. Because each OSM server interacts with a single Oracle RAC instance through the primary data source of its multi data source, all database operations for each order are directed to the same Oracle RAC instance. If ownership of an order is transferred to another OSM server (for example, when the cluster resizes or the order becomes a high-activity order), the processing of that order will be pinned again to the new OSM server.

Planning the Network Infrastructure

The following sections provide information about planning your network infrastructure.

Planning Network IP Addresses

The WebLogic Server cluster must have the following:

- A multicast IP address and port for the WebLogic Server cluster. Use any IP address between 224.0.0.0 and 239.255.255.255.
- IP addresses for each server in the cluster. If you are using whole server migration, the IP addresses must be available for node manager to dynamically allocate as floating IP addresses for the managed servers in the cluster.

The Oracle RAC database must have the following:

- Three IP addresses that resolve to the same SCAN host name
- Each Oracle RAC database instance must have a public and private IP address with corresponding host names.

Planning Bi-Directional Network and Firewall Access

Because JMS messages are transmitted in the context of JTA transactions, ensure that the WebLogic Server client always has bi-directional network and firewall access to every OSM WebLogic managed server. If you send a message to a distributed destination, for example, the JTA coordinator used by the WebLogic Server client must be able to communicate with

every managed server. Having only partial access to the managed servers in the OSM cluster can lead to inconsistent message states.

Network Latency Between WebLogic Server and the Database

To attain the fastest possible network connections, Oracle recommends that the physical servers for the WebLogic Server and Oracle Database be in the same network segment. The performance of OSM is sensitive to network latency between the WebLogic Server and the database.

Oracle recommends connecting the OSM and database servers with a minimal number of network devices in between. The switch connecting the network devices should be 10 GB capacity. This hardware configuration should produce an optimal network latency between 0.2 and 0.4 msec. Network latency above 1 msec can cause performance degradation.

Network Latency and NFS Configuration for WebLogic Server Shared Storage

The usual latency requirement from storage is service time of less than 5 msec. You must decide the IOPS (input/output per second) requirement based on hardware sizing.

For more information about recommended parameters for different levels of NFS mount robustness, see *Mount Options for Oracle files when used with NFS on NAS devices (Doc ID 359515.1)* on the Oracle support website at:

`https://support.oracle.com`

While this KM note was written for Oracle RAC, it provides a useful overview of various combinations of NFS parameters that are also appropriate for WebLogic Server shared storage.

Operating System Planning

Install OSM on UNIX or Linux systems for all uses - production, test, development, and demonstration. See *OSM Compatibility Matrix* for additional operating system requirements.

If you plan to use the Design Studio component of Oracle Communications Service Catalog and Design on a Windows system, you should download the SDK for your version of OSM and unzip it on the Windows system. If you plan to generate reports using the command line utility of the OSM Reporting Interface, download the SDK for your version of OSM and unzip it.

Database Planning

The following sections provide information about planning your database for the OSM system. In addition to the database planning information provided in this chapter, review the following:

- **Creating Tablespaces**: For more information about options and recommendations for creating tablespaces for OSM schemas, see "Tablespace and Schema Considerations for OSM Production Systems."
- **Using Partitioning**: For an overview of partitioning in OSM and a discussion about the benefits and pitfalls of partitioning, see *OSM System Administrator's Guide*. Oracle strongly recommends partitioning in all production deployments or production test environments, particularly those with high order volumes or any volume of large or

complex orders. Moreover, partitioning is required if you plan to use active-active Oracle RAC.

- **Order Purge Strategies**: For more information about order purge strategies, see *OSM System Administrator's Guide*. You must decide on an order purge strategy before doing performance testing and before going into production.
- **Sizing Partitions**: For more information about sizing partitions for order data, see *OSM System Administrator's Guide*. Partition sizing depends on your order purge strategy.
- **Cartridge Management Strategy**: For more information about cartridge management strategies, see *OSM System Administrator's Guide*.
- **Online and Offline Maintenance**: For more information about online and offline maintenance operations, see *OSM System Administrator's Guide*.
- **Database Management Procedures**: For more information about recommendations for managing your production database, see "Checking for Database Management Procedures."

Oracle RAC Database Active-Active Deployments

At the database-server layer, use Oracle RAC in the active-active high-availability topology for system test and production systems. In active-active Oracle RAC, all active instances simultaneously process database operations. In addition to load balancing, active-active Oracle RAC can also provide high availability if the physical database server of one Oracle RAC database instance is dimensioned to handle all of the database load upon failure of the other Oracle RAC database instance.OSM supports Oracle RAC through the use of WebLogic multi data sources. To optimize performance, OSM uses order affinity as described in the following section.

> **Note**
>
> OSM supports an Oracle RAC configuration of two or more nodes. OSM also supports Oracle RAC One Node.

Database Partitioning

During the installation, you specify if you need to partition the OSM database, and you provide partition sizes. Oracle strongly recommends using partitioning for production databases and production test databases.

You can change the values that you selected during the installation process. However, those updates do not affect existing partition sizes.

For information about partition sizes, see *OSM System Administrator's Guide*.

Database Failover with Oracle RAC

In an Oracle RAC configuration, all data files, control files, and parameter files are shared for use by all Oracle RAC instances. When a database instance fails, performance may be temporarily affected, but the database service continues to be available.

The use of WebLogic multi data source and JMS minimize the impact of a database failure in the following ways:

- The JDBC multi data source fails over to the secondary data source.
- In-flight WebLogic transactions are driven to completion or rolled back, based on the state of the transaction at the time of the failure. Because JMS messages are redelivered, most failed transactions are automatically retried (upon redelivery) on the secondary data source. This does not apply to failed web services and HTTP requests (for example, failed createOrder requests must be resubmitted).

Database Failover with Oracle RAC One Node

For Oracle RAC One Node, there is only one instance active at a time. Therefore, a stand-alone data source using the SCAN address (without Instance Name) ensures that all OSM managed servers communicate with the same database instance while still allowing for automated failover.

Install OSM as if using a non-RAC DB using the SCAN address (without instance name). OSM treats an Oracle RAC One Node database as if it were a non-RAC database and lets the database and SCAN listener handle failover.

For RAC One Node, a sample data source URL is:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host1)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=OSM)))
```

Notice that INSTANCE_NAME is not provided, because the SCAN listener chooses the instance when a connection is made based on the current failover state of the database.

> **Note**
>
> The considerations described in "Listener Considerations for Oracle RAC" are not applicable to Oracle RAC One Node.

Listener Considerations for Oracle RAC

In an Oracle RAC environment, the database listener process establishes the connections between the JDBC data source of a WebLogic Server instance and an Oracle RAC instance.

To enable the listener functionality, Oracle recommends that you use remote listeners. Remote listeners are also known as SCAN listeners. With this option, each Oracle RAC instance is configured to register with a remote listener that may or may not be in the same physical server. There is no "remote listener only" scenario: local listeners must be running for the remote listener to work properly. When a request comes in, the remote listener redirects it to the local listener depending on what instance it is connecting to.

When configuring JDBC data sources, you must be aware of your listener process setup. The OSM installer will automatically configure your JDBC data sources based on the listener process considerations discussed in the following sections. These considerations apply to both active-active and active-passive topologies.

Remote Listener Considerations

By default, server-side load balancing is configured when using remote listeners. That is, the remote listener decides how to forward connection requests based on the load of the Oracle RAC instances. OSM active-active configurations require that server-side load balancing be

overridden. To achieve this, the OSM installer includes the INSTANCE_NAME parameter (the SID of a specific instance) in the JDBC URL of each member data source, in addition to identifying the database service by name.

For example, the following data source URLs include both INSTANCE_NAME and SERVICE_NAME:

```
jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host1)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=OSM) (INSTANCE_NAME=SID1)))

jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=host1)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=OSM) (INSTANCE_NAME=SID2)))
```

In the example, the host and port are for the SCAN listener, the service name is the same, and the instance names are different.

The OSM installer will automatically set up the URL of each JDBC data source in the WebLogic Server instances' multi data source. However, if you choose to manually configure additional Oracle RAC instances, you must populate the SID in the JDBC URL of the member data sources in WebLogic Server. See "Manually Configuring Additional Data Sources for an Oracle RAC Instance" for details.

Local Listener Considerations

When configuring local listeners, consider the following:

- Each database instance should be configured to register only with its local listener.
- Oracle instances can be configured to register with the listener statically in the **listener.ora** file or registered dynamically using the instance initialization parameter **local_listener**, or both. Oracle recommends using dynamic registration.
- A listener can start either a shared dispatcher process or a dedicated process. Oracle recommends using dedicated processes.

WebLogic Server Planning

An OSM instance consists of an administration server, and a cluster of managed servers. Clustering ensures continuous availability of your OSM server and improves performance by enabling load balancing, scalability, and failover. You may choose to use the clustering feature in OSM if:

- You want to minimize unexpected system downtime.
- Your order volume is very high and cannot be sustained with a single WebLogic Server instance or physical host.

OSM supports the following load balancing:

- **Load balancing for JMS messages**: The native WebLogic load balancing options for Java Messaging Service (JMS) messages help OSM maximize server resource use and order throughput. Load balancing also enables OSM to minimize server response time and processing delays that can occur if some servers are overloaded with orders while others remain unused. Load balancing allows rolling downtimes of servers without client impact, as long as enough servers are up and running.
- **Load balancing for HTTP and HTTPS messages**: In addition to the native WebLogic support for load balancing JMS messages, Oracle recommends installing a software or hardware HTTP load balancer for balancing incoming HTTP or HTTPS messages.

To ensure high availability, the load balancing mechanisms (both the native WebLogic JMS load balancing or the HTTP load balancer) forward messages to other managed servers if one of the managed servers fails. Orders that were being processed by the failed server are delayed until that server is either restarted or migrated.

Understanding the WebLogic Cluster Configuration

Recommendations for OSM cluster include the following:

- **Messaging mode**: Oracle recommends that you use multicast messaging mode when setting up OSM. For more information about using multicast or unicast, see "About the WebLogic Messaging Mode and OSM Cluster Size."
- **Load balancing**: Set up clusters to use the random algorithm.

Table 3-5 includes a summary of the recommendations for OSM cluster.

Table 3-5 Configuration Recommendations for OSM Cluster

Configuration Item	Value
Cluster Messaging Mode	multicast
Default Load Algorithm	Random

About Cluster Domain Management

Oracle recommends the following best practices when configuring managed server instances in your clustered OSM domain.

- Configure Node Manager to automatically restart all managed servers in the domain.
- Configure all managed server instances to use MSI, which is the default. This feature allows the managed servers to restart even if the administration server is unreachable due to a network, hardware, or software failure. See *Oracle Fusion Middleware Managing Server Startup and Shutdown for Oracle WebLogic Server* for more information.

About the WebLogic Messaging Mode and OSM Cluster Size

The WebLogic Server cluster messaging mode enables cluster members to remain synchronized and provides the foundation for other WebLogic Server functions such as load balancing, scalability, and high availability.

In an OSM cluster, the messaging mode can be multicast or unicast. Oracle recommends using multicast in most OSM installations because multicast is generally more reliable.

In some cases, unicast may be the only option. For example, multicast can only work over a single subnet. If a single subnet is not possible due to technological or IT policy reasons, or if the network's multicast transmission is not reliable, then unicast messaging mode becomes the best option. If you must use unicast, ensure that you apply the WebLogic Server patches that resolve the currently known unicast issues. See "Software Requirements" for patch information.

Do not use unicast for a cluster with more than 20 managed servers. Enabling a reliable multicast network for WebLogic Server multicast messaging mode is the only option for such large cluster sizes. The broadcast nature of multicast over UDP packets works better in such large clusters than one-to-one TCP unicast connections between each pair of managed servers.

You can use unicast for cluster sizes between 10 and 20 managed servers, but consider multicast if you begin to experience poor performance or reliability issues.

About Coherence and Unicast

Oracle recommends unicast mode for Oracle Coherence. The OSM cluster performance and robustness are sensitive to the synchronization of cached data maintained by Coherence. The inherently unreliable packet delivery with UDP in multicast transmission may destabilize cache synchronization, and errors can be difficult to troubleshoot. As a result, Oracle does not recommend using Coherence in multicast mode.

Understanding the Administration Server

The administration server operates as the central control entity for the configuration of your OSM WebLogic domain.

The failure of the administration server does not affect the operation of managed servers in the domain. Furthermore, the load balancing and failover capabilities supported by the domain configuration remain available. However, it does prevent you from changing the domain's configuration, including loss of in-progress management and deployment operations and loss of ongoing logging functionality.

Oracle recommends the following best practices when configuring the administration server in your OSM WebLogic domain:

- The administration server should not participate in a cluster. Ensure that the administration server's IP address is not included in the cluster-wide DNS name.
- Start the administration server using Node Manager to ensure that the administration server restarts in the event of a failure. (If the administration server for a domain becomes unavailable, the managed servers in the domain will periodically attempt to reconnect to the administration server.) Dot not deploy OSM to the administration server in a production system.

For additional clustering best practices, see *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*.

You may also consider transforming the administration server in an existing domain for cold cluster failover.

In this active-passive topology, the administration server is installed on Node1 and then transformed to a shared disk. In the event of failure, it will be failed over to Node2. The administration server *domain_home* resides on a shared disk that is mountable by both Node1 and Node2 but is mounted by either one of the two at any given point in time. The listen address of the administration server is a virtual IP.

See the chapter on active-passive topologies in *Oracle Fusion Middleware High Availability Guide* for full details.

Understanding Node Manager Configuration

Node Manager is a Java utility that runs as a separate process from the WebLogic Server and allows you to perform common operations for a managed server, regardless of its location with respect to its administration server. The Node Manager process is associated with a machine. Thus each physical server has its own Node Manager, which can control all server instances that reside on the same machine as the Node Manager process.

Consider the following guidelines when using Node Manager:

- Run Node Manager as an operating system service on UNIX platforms, allowing it to restart automatically when the system is restarted.
- Set the **AutoRestart** attribute of the administration server and each managed server to **true** to allow Node Manager to automatically restart it in the event of failure, depending on the exit code (if the exit code is less than 0, the server is not restarted and you must diagnose the problem).
- Do not disable the Managed Server Independence (MSI) mode for a managed server (enabled by default). MSI allows Node Manager to automatically restart a managed server after failure even when the administration server is unavailable.
- To ensure that Node Manager properly restarts servers after a system crash (for example, an operating system crash), you must do the following:
  - Ensure that **CrashRecoveryEnabled** is set to **true**. This property is disabled by default.
  - Start the administration server using Node Manager. You cannot use Node Manager to start a server instance in MSI mode, only to restart it. For a routine startup, Node Manager requires access to the administration server.
  - Start all managed servers using the administration server. You can accomplish this using the WebLogic Server Scripting Tool command line or scripts or the Remote Console. A widespread practice is to start managed servers using a shell script.

See *Node Manager Administrator's Guide for Oracle WebLogic Server* for more information.

Understanding JMS Messaging

Recommendations for JMS Messaging include the following:

- For production systems, Oracle recommends that JMS message persistence be configured to use a JDBC store rather than a file store. WebLogic added support for the use of JDBC stores for tlogs in Fusion MiddleWare 12cR1. From an operational perspective, creating JDBC stores for JMS and tlogs on the same database instance used for the OSM schema allows for atomic backups thereby enabling consistent restoration of overall application state. Note that the use of JDBC stores increases database CPU utilization and storage I/O operations; this needs to be accounted for in infrastructure planning. The exact magnitude of the cost increase varies depending on infrastructure and solution characteristics.

  For optimal performance, for systems using a file store, Oracle recommends that you configure Direct-Write-With-Cache, if this option is supported in your environment. For information about the best practices for configuring a WebLogic file store, see the chapter about using the WebLogic persistent store in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*.
- In a clustered environment, WebLogic uses load balancing to distribute the workload across clusters. For OSM, set the load balancing policy for distributed JMS queues to **Random**. For more information about WebLogic JMS Random distribution, see the chapter about configuring advanced JMS system resources in *Oracle Fusion Middleware Configuring and Managing JMS for Oracle WebLogic Server*.
- WebLogic supports the Store and Forward (SAF) service for reliable delivery of messages between distributed applications running on different WebLogic Server instances. It is recommended that you set the Conversation Idle Time Maximum on SAF agents to a positive value to allow messages to be forwarded to other active members when the original target is down or unavailable. For more information about the WebLogic SAF service, see the chapter about understanding the SAF service in *Oracle Fusion Middleware Configuring and Managing Store-and-Forward for Oracle WebLogic Server*.

Oracle recommends that you use SAF to integrate OSM with Oracle Communications ASAP, Oracle Communications IP Service Activator, and Oracle Communications Unified Inventory Management (UIM). For more information about this post OSM installation task, see "OSM Integration with External Systems."

JMS Distributed Destinations

JMS destinations, which may be JMS queues or JMS topics, serve as repositories for messages. A JMS destination provides a specific end point for messages, which a JMS client uses to specify the target of messages that it produces and the source of messages that it consumes. For example, OSM automation plug-ins can specify the JNDI names of the JMS queue and the JMS reply-to queue to produce and consume messages with external systems.

A distributed destination is a single set of destinations that are accessible as a single, logical destination to a client (for example, a distributed topic has its own JNDI name). The members of the set are typically distributed across multiple servers within a cluster, with each member belonging to a separate JMS server. When deployed to a cluster, OSM uses distributed destinations because JMS provides load balancing and failover for the members of a distributed destination in a cluster. For performance reasons, the server affinity is enabled on the connection factory to give preference to local destination members.

> **Note**
>
> OSM does not support uniform distributed destinations (UDDs), which are the default type of distributed destination in WebLogic. OSM supports only weighted distributed destinations (WDDs). When configuring distributed destinations in the WebLogic Server Administration Console, select **Weighted** for **Destination Type** to configure the distributed destination as a WDD.

> **Note**
>
> Messages sent to JMS distributed destinations will always be delivered to member queues. However, messages delivered to a member queue can get stuck in the event of a server failure. In that case, messages cannot be consumed until either the WebLogic server is restarted or the JMS server is migrated.

Cluster and Single-Server Queues

Multiple queues are created automatically when OSM is installed. When OSM is installed to a cluster, additional queues are provided for added efficiency in processing when OSM is in a WebLogic cluster.

If your development systems have been installed onto a single-server instance of WebLogic Server, ensure that your client systems are updated to use the queues appropriate to a cluster. For more information about the queues that are installed with OSM, see the discussion of OSM installed components in *OSM System Administrator's Guide*.

About WebLogic Server JMS T3 and T3S Load Balancing

WebLogic Server T3 and the secure T3S variant are transport protocols for application-level services that OSM uses for communication between client applications and the WebLogic

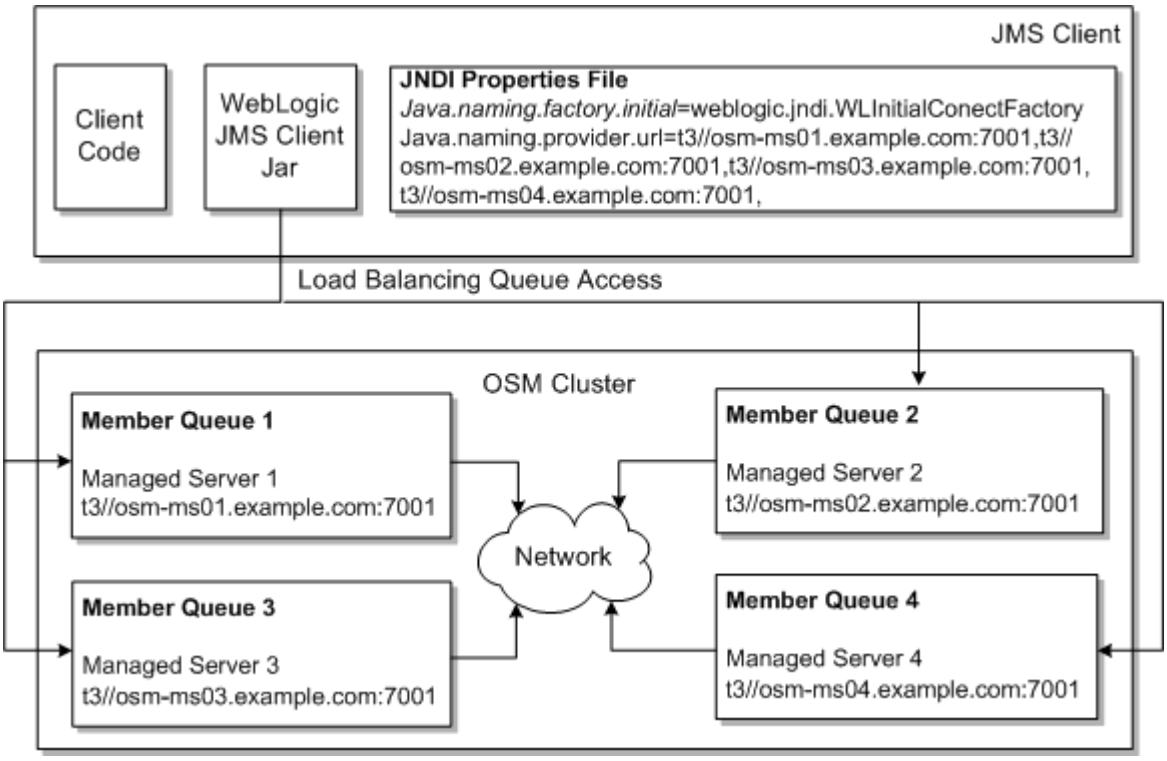
Server. OSM typically communicates messages using T3 or T3S to distributed JMS destination for:

- OSM Web Service XML/SOAP messages, like an OSM CreateOrder Web Service request from a CRM that initiates an OSM order
- OSM automations that receive messages from external fulfillment systems
- OSM internal event handling. For example, **oms_events_queue** can be used for triggering data change notifications on an order.

The T3 protocol is fully implemented by the WebLogic Server and its client libraries. T3 client libraries support clustered server URLs for the initial context lookup enabling native WebLogic support for load balancing. The WebLogic Server cluster then manages load distribution based on system availability and the selected load balancing schema. See the WebLogic documentation for more information about creating a WebLogic client for communicating JMS messages to OSM over T3.

Figure 3-2 shows a JMS client with a JNDI properties file configured with the URLs for each managed server within a cluster that the client is sending messages to. WebLogic Server load balances these messages using the URLs based on a load balancing schema (see "About JMS Load Balancing Schema Options" for more information). OSM typically processes an OSM order on the managed server that receives the order, but in some cases OSM uses the managed server queues internally to redistribute orders to other managed servers after an order has been received on a managed server (see "About Order Affinity and Ownership in an OSM WebLogic Cluster" for more information).

Figure 3-2 JMS Load Balancing



About JMS Load Balancing Schema Options

At the WebLogic Server cluster level, you can select between three load balancing schemas for JMS load balancing. WebLogic supports only one load balancing schema selection per cluster even if the cluster hosts multiple applications. The load balancing schema you select effects OSM both on its external and internal messaging interfaces such as incoming messages from an external system or messages exchanged between managed servers within the cluster.

The following lists the WebLogic Server load balancing options:

- Round-robin: OSM distributes the JMS messages evenly across the managed servers in the cluster by periodically circulating the list of available managed servers. Each server is treated equally.
- Random-based: Before routing a JMS message, the random-based load balancing schema generates a random number and selects one of the candidate servers as the message destination based on the random number.
- Weight-based: If the OSM cluster consists of managed servers hosted on systems with varying hardware resource capacity, you can assign load balancing weights to each WebLogic Server instance.

Oracle recommends that you use random-based load balancing. The OSM Installer automatically configures random based load balancing.

Understanding Whole Server Migration for High Availability

The OSM automation framework uses WebLogic Java Message Service (JMS) to support messaging between automation components and external systems. In addition, upstream systems can access OSM web services with JMS as one of the transport protocols. Thus, it is critical that JMS be highly available. JMS high availability is achieved by using JMS distributed destinations (see "JMS Distributed Destinations") as well as planning for whole server migration in the event of failure.

WebLogic Server migration is the process of moving a managed server instance elsewhere in the event of failure. In the case of whole server migration, the server instance is migrated to a different physical machine upon failure. Whole server migration is the preferred and recommended approach because all JMS-related services are migrated together.

> **Note**
>
> JMS service migration is not supported with OSM, because using JMS service migration could result in the JMS server not running on the same machine as the managed server to which it is dedicated. This would cause issues similar to those that would arise if server affinity was not configured on the default OSM JMS connection factory.

WebLogic Server provides migratable servers to make JMS and the JTA transaction system highly available. Migratable servers (clustered server instances that migrate to target servers) provide for both automatic and manual migration at the server level, rather than at the service level.

When a migratable server becomes unavailable (for example, if it hangs, loses network connectivity, or its host machine fails), migration is automatic. Upon failure, a migratable server

is automatically restarted on the same machine if possible. If the migratable server cannot be restarted on the machine where it failed, it is migrated to another machine. In addition, an administrator can manually initiate migration of a server instance.

The target server for the migration can be a spare server on which Node Manager is running. This server does not participate in the cluster until a migratable server is migrated to it.

Another option for the target server is a server that is hosting a WebLogic Server instance. In the event of failure, the migratable server will be migrated to it, resulting in the two instances (which now run on the same server) competing for CPU, memory, and disk resources. In this case, performance could be impacted.

Before you configure automatic whole server migration, be aware of the following requirements:

- All servers hosting migratable servers are time-synchronized. Although migration works when servers are not time-synchronized, time-synchronized servers are recommended in a clustered environment.
- To ensure file availability, use a disk that is accessible from all machines. If you cannot share disks between servers, you must ensure that the contents of *domain_home***/bin** are copied to each machine.
- Ensure that the user account that runs the managed servers can work without a password prompt.
- Ensure that the user account that runs the managed servers have execute privilege on the **/sbin/ifconfig** and **/sbin/arping** binaries that are involved in creating floating IP address.
- Use high-availability storage for state data. For highest reliability, use a shared storage solution that is itself highly available; for example, a storage area network (SAN). For more information, see *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*.
- For capacity planning in a production environment, keep in mind that server startup during migration taxes CPU utilization. You cannot assume that because a machine can handle a certain number of servers running concurrently that it also can handle that same number of servers starting up on the same machine at the same time.

For additional requirements, see *Oracle Fusion Middleware Using Clusters for Oracle WebLogic Server*.

Managing WebLogic Transactions

Transactions are a means to guarantee that database changes are completed accurately. The WebLogic Server transaction manager is designed to recover from system crashes with minimal user intervention and makes every effort to resolve transaction branches with a commit or roll back, even after multiple crashes or crashes during recovery.

To facilitate recovery after a crash, the WebLogic Server Transaction Recovery Service automatically attempts to recover transactions on system startup. On startup, the Transaction Recovery Service parses all transaction log records for incomplete transactions and completes them as described in *Oracle Fusion Middleware Programming JTA for Oracle WebLogic Server*.

Oracle recommends the following guidelines:

- If a server crashes and you do not expect to be able to restart it within a reasonable period of time, you can migrate either the whole server or the Transaction Recovery Service to another server in the same cluster. The transaction log records are stored in the default persistent store for the server. If the default persistent store is a file store (the default), it

must reside in a shared storage system that is accessible to any potential machine to which a failed migratable server might be migrated. See "Persistent Store: JMS File Store and JDBC Store" for high-availability considerations.

- Configure server instances using DNS names rather than IP addresses. A server instance is identified by its URL (IP address or DNS name plus the listening port number). Changing the URL by moving the server to a new machine or changing its listening port on the same machine effectively moves the server, so the server identity may no longer match the information stored in the transaction logs. Consequently, *any pending transactions stored in the transaction log files will be unrecoverable*. This is also critical if firewalls are used to avoid address translation issues.
- First, attempt to restart a crashed server and allow the Transaction Recovery Service to handle incomplete transactions (rather than move it to a new machine). However, if the server exited with a code less than 0, do not attempt to restart it unless you diagnose the problem. In this case, the server did not terminate in a stable condition; for example, due to invalid configuration.

  See *Oracle Fusion Middleware Programming JTA for Oracle WebLogic Server* for more information.

Persistent Store: JMS File Store and JDBC Store

WebLogic's persistent store provides a built-in, high-performance storage solution for WebLogic Server subsystems and services that require persistence. For example, it can store persistent JMS messages or temporarily store messages sent using the Store-and-Forward feature. The persistent store supports persistence to a file-based store or to a JDBC-enabled database. The persistent store is important for OSM because it stores all of the JMS messages from the JMS service.

There is a trade-off in performance and ease of backup when choosing between JMS file store and JDBC store:

- JMS file store provides better performance than JDBC store. However, you cannot perform an online backup of OSM data consistent with the JMS file store. You must first shut down OSM and then back up the JMS file store and the database at the same time. Otherwise, inconsistent message states and database states may result, and the backup cannot be used to restore OSM.
- The benefit of JDBC store is that online database backups can obtain consistent snapshots of both OSM data and JMS messages. For more information, see *Persistent Store Configuration & Operational Considerations for JMS, SAF & WebLogic tlogs in OSM* [Doc ID: 2469767.1] knowledge article on the Oracle support website at: https://support.oracle.com.
- In an Oracle Communications environment where ASAP, IP Service Activator, or UIM is running in the same OSM WebLogic domain, the JDBC store may yield a more consistent backup strategy across the domain and may outweigh performance considerations. However, you cannot take a consistent backup of OSM because the data is distributed across the database and file system.

To realize high availability for the file store, it should reside on shared disk storage that is itself highly available, for example, a storage area network (SAN).

If you choose JMS file store, Oracle recommends that you configure one custom file store for each managed server.

Persistent Store: TLog File Store and JDBC Store

Each managed server is associated with a transaction log (TLog) store. For production OSM systems, Oracle recommends replacing the Default Store, which is a file-based store, with a JDBC store. In a RAC environment, TLog JDBC stores can share a common multidata source configured with load balancing or a common GridLink data source if this option is licensed in your environment.

For more details about using a JDBC TLog Store, see the chapter about using a JDBC Store in *Oracle Fusion Middleware Administering the WebLogic Persistent Store*.

Understanding Hardware or Software HTTP and HTTPS Load Balancing Options

Oracle recommends that you use an HTTP Server to load balance HTTP and HTTPS messages for OSM clusters in production environments that require high availability for HTTP messages (for example, for the OSM web clients or for OSM messages over HTTP).

> **Note**
>
> These recommendations only apply to HTTP and HTTPS messages. JMS messages should be load balanced using the WebLogic Server native support for JMS T3 and T3S load balancing (see "About WebLogic Server JMS T3 and T3S Load Balancing").

Oracle recommends using external load-balancing solutions, such as Oracle HTTP Server or Apache HTTP Server.

> **Note**
>
> The HTTP proxy that is managed by the WebLogic Administration Console has been deprecated starting from FMW 14.1.2. Refer to Deprecated and Removed Functionality in *What's New in Oracle Weblogic Server* for more information.

You can also consider a hardware load-balancing solution for load balancing HTTP and HTTPS messages. A hardware load balancer can use any algorithm supported by the hardware, including advanced load-based balancing strategies that monitor the utilization of individual machines. If you choose to use hardware to load balance HTTP and HTTPS sessions, the hardware must support a compatible passive or active cookie persistence mechanism and SSL persistence.

The following lists possible software load-balancer options for OSM HTTP and HTTPS messages:

- Oracle HTTP Server
- Oracle WebLogic Server proxy plug-ins for the following standard web server solutions
  - Oracle iPlanet Web Server (7.0.9 and later)
  - Apache HTTPD 2.2.x
  - Microsoft Internet Information Services 6.0 and 7.0

- Dedicated software load-balancing solutions like Oracle Traffic Director. Oracle recommends this option for running OSM with Oracle Exalogic and Oracle SuperCluster.

The following lists possible hardware load-balancer options for OSM HTTP and HTTPS messages:

- F5 Big-IP
- Cisco ACE

About HTTP and HTTPS Load Balancing and Session ID Configuration

Round-robin load balancing for HTTP and HTTPS messages is the only supported option for software load balancers because WebLogic does not propagate managed server weights externally.

> **Note**
>
> The WebLogic cluster load balancing schema options (see "About JMS Load Balancing Schema Options") have no effect for load balancing HTTP messages because an HTTP load balancer is outside of the cluster.

When running OSM in a cluster, you must enable the proxy-plug-in option at the cluster level as opposed to the managed-server level, otherwise session drops may occur and login to the OSM web clients may not be possible.

You must ensure that the chosen HTTP load-balancing solution supports WebLogic session IDs and custom HTTP headers. All WebLogic plug-ins, including the Oracle HTTP Server, support sticky sessions, but they do not support session ID failover if the server the ID is connected to fails.

About Oracle Coherence

Oracle Coherence plays a major role in providing grid services to OSM. OSM employs the Coherence Invocation service as well as local and distributed Coherence caches. The Coherence Invocation service is a feature of the Coherence Application Edition or Grid Edition.

Oracle Coherence must be configured to avoid conflicts in a clustered OSM environment. See the WebLogic documentation for guidelines and best practices.

4

Installing and Configuring the Oracle RAC Database

This chapter provides information about the installation and configuration of Oracle Real Application Cluster (Oracle RAC) Database that is specific to Oracle Communications Order and Service Management (OSM).

For information about installing OSM on Oracle RAC One Node, see "Database Failover with Oracle RAC One Node".

For complete installation instructions and general information about installing and configuring the Oracle Database, see the Oracle Database documentation.

Database Information You Should Record

Some of the information that you set when installing the Oracle Database will be needed during the OSM installation. Record the following information and provide it to the OSM installer:

- Oracle RAC database instance hosts
- Oracle RAC database instance ports
- Oracle RAC database instance SIDs
- Oracle RAC database service name
- Database Administrator User name/Password
- Information about all tablespaces created for OSM

Creating the Oracle Database for OSM

This section describes how to create and configure an Oracle database for OSM. It also gives installation and configuration guidelines to improve OSM performance. Although these guidelines can help improve OSM performance, the hardware and configuration used for Oracle Server running the OSM database schema have the largest impact on performance.

Some of the procedures in this section must be performed by an Oracle Database Administrator (DBA).

The following database user roles and permissions are required:

```
GRANT CREATE ANY CONTEXT TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN
OPTION;
GRANT QUERY REWRITE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN
OPTION;
GRANT CREATE TABLE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN
OPTION;
GRANT GRANT ANY PRIVILEGE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
ADMIN OPTION;
GRANT CREATE USER TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN OPTION;
```

```
GRANT CREATE ANY VIEW TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN
OPTION;
GRANT UNLIMITED TABLESPACE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
ADMIN OPTION;
GRANT CREATE MATERIALIZED VIEW TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
ADMIN OPTION;
GRANT CREATE SYNONYM TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN
OPTION;
GRANT SELECT ON SYS.V_$PARAMETER TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_
WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_TABLESPACES TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_
WITH GRANT OPTION;
GRANT SELECT ON SYS.DBA_JOBS TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
GRANT OPTION;
GRANT SELECT ON SYS.DBA_AUTOTASK_CLIENT_JOB TO
_REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH GRANT OPTION;
GRANT EXECUTE ON SYS.DBMS_LOB TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
GRANT OPTION;
GRANT EXECUTE ON SYS.DBMS_LOCK TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
GRANT OPTION;
GRANT EXECUTE ON SYS.UTL_FILE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
GRANT OPTION;
GRANT RESOURCE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN OPTION;
GRANT IMP_FULL_DATABASE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN
OPTION;
GRANT CONNECT TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN OPTION;
GRANT DBA TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN OPTION;
GRANT EXP_FULL_DATABASE TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH ADMIN
OPTION;
GRANT EXECUTE ON DBMS_RANDOM TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
GRANT OPTION;
GRANT EXECUTE ON SYS.DBMS_SCHEDULER TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_
WITH GRANT OPTION;
GRANT EXECUTE ON UTL_HTTP TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
GRANT OPTION;
GRANT EXECUTE ON DBMS_SQL TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH
GRANT OPTION;
GRANT EXECUTE ON UTL_TCP TO _REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_ WITH GRANT
OPTION;
```

where *_REPLACE_THIS_TEXT_WITH_DB_ADMIN_NAME_* is the account used by the DBA to install OSM.

You can install OSM using Oracle Database pluggable databases (PDB) within a multitenant container database (CDB) or using a non-container database.

A higher level of availability and performance is offered by Oracle Real Application Clusters (Oracle RAC). For OSM production environments, Oracle recommends Oracle RAC in the active-active mode. The data files that comprise the database must reside on shared storage that is accessible to all servers in the database cluster. If a node in the cluster fails, Oracle Database continues to run on the remaining nodes.

OSM supports Oracle RAC through WebLogic multi data sources. For each WebLogic managed server, one database instance is configured as the primary instance (in the WebLogic multi data source) and the others as secondary instances. WebLogic monitors the status of the primary instance and fails over to a secondary instance in the event of failure.

Setting Up the Database and Clusterware for Oracle RAC

The following procedure describes how to set up Oracle Database and Clusterware to support Oracle RAC.

1. Install Oracle Clusterware on all nodes:
   - For Linux, see *Oracle Grid Infrastructure Installation Guide for Linux*.
2. Install Oracle Database with Real Application Clusters on all nodes.

   For Linux and UNIX, see *Oracle Real Application Clusters Installation Guide for Linux and UNIX*.
3. Create an ASM instance and disk group with shared storage accessible to all nodes.
4. Use Database Configuration Assistant (DBCA) to create an Oracle RAC database on all nodes, using the ASM disk group created in the previous step as storage.
5. Create virtual IP addresses for the database instances, and one Single Client Access Name (SCAN) for the database cluster.

   In an Oracle RAC environment, remote listeners are typically used. When you create the database, the remote listener is created as a SCAN listener. The SCAN resolves to multiple IP addresses in the cluster.
6. Start the database and listeners on all nodes.
7. Download and install database patches. See "Software Requirements" for information about the patches needed for the database for your platform. Ensure that all of the patches have been downloaded from Oracle support and installed before installing OSM.
8. For OSM, you also need to add a permission to the database administrator user that you are going to use during the OSM installation. To do this, log in to SQL*Plus as sysdba and run the following:

   ```
   grant create any context to sysuser as sysdba with admin option
   ```

   where *sysuser* is a user with sysdba privileges that you intend to use during OSM installation.

You have now configured your database instances to support Oracle RAC and OSM. When you run the OSM installer, you have the option to create the WebLogic multi data source and data sources required for this configuration. See "Installing OSM" for more information.

Memory Settings for the OSM Database

It is a good idea to allocate as much memory as possible to the database. Use memory guidelines provided in "Planning the Physical Architecture" then confirm your memory requirements with performance testing and tuning.

Oracle recommends the use of Automatic Shared Memory Management (ASMM) to manage the memory on your system in production environments. ASMM is a better option than Automatic Memory Management (AMM) because AMM can cause performance problems.

If you are using Linux for your database server, and your database is using approximately 32 GB of memory or more, you may want to implement Linux HugePages for memory performance improvements. Linux HugePages is incompatible with AMM. For more information about HugePages, consult your operating system documentation.

Character Sets

Oracle recommends using the AL32UTF8 character set for the OSM database instance. However, if OSM is the only application that will be using the database instance, you can use the default character set for your location.

The national character set can be left to the default value for your location.

Database Parameters

This section outlines suggested Relational Database Management System (RDBMS) server configurations for OSM.

> **Note**
>
> This section provides suggested values for use with OSM in a production system. The suggested values are guidelines only. The values you use will depend on your system type and actual processing requirements.

Table 4-1 shows the parameters that should be set for all databases. When using a multi-tenant container database (CDB), refer to the Level column to determine if the parameter should be set on the CDB or a PDB.

Table 4-1 Suggested Oracle Database Parameters for All Systems

Parameter	Level	Value and Description
awr_pdb_autoflush_enabled	CDB	TRUE This is required to enable AWR snapshots at the PDB level. By default, these snapshots are disabled.
db_files	CDB	16384 The maximum number of database files that can be opened for the database.
db_writer_processes	CDB	4 (SPARC T3 and T4 only) If you are using a SPARC T3 or T4, the default value of this parameter can cause severe performance degradation.
distributed_lock_timeout	CDB	This value should be greater than the value specified for WebLogic Server domain Java Transaction Timeout. It should also be greater than the value specified for OSM cartridge deployment timeout. You set this value when you create the WebLogic Server cluster. For more information, see "Preventing Connection Timeout when Using a Remote Database." Setting this value to a value greater than the JTA value avoids situations where in a database table lock's timeout expires prior to WebLogic Server JTA or JDBC XA transaction timeout. This can result in ORA-02049 errors ("timeout: distributed transaction waiting for lock"). For more information, refer to *Parameters That Help Prevent Timeouts from Occurring on the OSM Server for both Deployment and Undeployment Operations* (Doc ID 2187032.1). knowledge article on My Oracle Support.

Table 4-1 (Cont.) Suggested Oracle Database Parameters for All Systems

Parameter	Level	Value and Description
session_cached_cursors	CBD	100 This helps reduce parsing by increasing the number of cursors cached for each database session.
session_max_open_files	CBD	50 The maximum number of BFILEs that can be opened.
cursor_sharing	PDB	FORCE This parameter must be set to FORCE to avoid library cache lock, which is a result of scenarios where in there is a large number of statements in the shared pool that differ only in the values of their literals, and database response time is low due to a very high number of library cache misses (for example, because of hard parses and library cache latch contention).
deferred_segment_creation	PDB	FALSE In high volume deployments, especially on Oracle RAC, deferred segment creation can lead to serious performance issues when the database is forced to create the deferred segments of a partition in order to store new orders. This occurs when the previous partition is exhausted. The result is high “library cache lock” waits that could last for an extended period of time (frequently, more than 30 minutes).
O7_DICTIONARY_ACCESSIBILITY	PDB	FALSE (default) You can set this parameter to TRUE before running the OSM installer, if you are planning to use **sys** as the database administrator user to use during the OSM installation. If you do this, you should set the option back to the default of FALSE, after the installation of OSM is complete. If you leave this parameter to FALSE, you must append **sysdba** to the user name when entering the database administrator credentials in the OSM installer.
open_cursors	PDB	2000 Maximum number of open cursors a session can have at once.
optimizer_mode	PDB	ALL_ROWS (default) With ALL ROWS, the optimizer uses a cost-based approach for all SQL statements in the session and optimizes with a goal of best throughput (minimum resource use to complete the entire statement).
parallel_degree_policy	PDB	AUTO AUTO enables automatic degree of parallelism, statement queuing, and in-memory parallel execution.
_optimizer_invalidation_period	PDB	600 Otherwise, it may take 3 hours, instead of 10 minutes to benefit from new database optimizer statistics.

Oracle recommends that you do not use the Shared Server configuration (called Multi-Threaded Server, or MTS in previous versions of Oracle Database) for production systems running OSM, for performance reasons. OSM implements its own connection multiplexing.

Configuring Time Zone Settings in the Database

The database server running OSM must not use Daylight Savings Time (DST); otherwise date and schedule calculations will be incorrect during Daylight Savings Time. You can avoid this problem in the following ways:

- Set the time zone of the database to UTC (Coordinated Universal Time, formerly Greenwich Mean Time)
- Set the time zone of the database as an offset to UTC, in the format +/-hh:mm.

Ensure that the operating system time zone setting of the user starting the database processes is set using the considerations above.

Preventing Stuck Orders Due to Inactive Database Sessions

When a machine on which OSM is running is abruptly shut down, OSM orders may become stuck due to inactive database sessions that may not be cleaned for an extended period of time. When this happens, locks associated with these sessions are not released.

To avoid this problem, do the following:

1. Configure your database machine so that these inactive database sessions are automatically killed after 10 minutes. Do the following:
   - Add **enable=broken** in **tnsnames.ora**.
   - Add **SQLNET.EXPIRE_TIME=10** in **sqlnet.ora**.
2. On a Linux system, as root, configure TCP keepalive as follows:

```
sysctl -w net.ipv4.tcp_keepalive_time=600
sysctl -w net.ipv4.tcp_keepalive_intvl=60
```

> **Note**
>
> This change can be made permanent by adding the following lines to **/etc/sysctl.conf**:
>
> ```
> net.ipv4.tcp_keepalive_time=600
> net.ipv4.tcp_keepalive_intvl=60
> ```

Tablespace and Schema Considerations for OSM Production Systems

This section contains information and settings for use when creating any instance of the Oracle database to be used by OSM, including production and development systems.

If you are creating the database from the Database Configuration Assistant, Oracle recommends that you use the "Custom Database" template.

Sizing the OSM Database Schemas

The OSM installer creates the following schemas:

- The core schema, which contains order cartridge metadata, order data, configuration, and other data.
- The rule engine schema, which contains logic for rule processing.
- The reporting schema, which is used for reporting.

The sizing of production systems is multi-dimensional and dependent on many variables that vary greatly from customer to customer, such as daily transaction volume and amount of historical data to be maintained.

For help determining the sizing of your production system, contact Oracle Support.

OSM allows you to add additional partitions to store order data as needed. This means that as orders are entered into the system and the available storage is used, additional partitions on new tablespaces can be added to your environment. The space needed for these additional partitions does not need to be calculated at installation time. Orders can also be purged from the system based on the partition they are in.

For the initial sizing details about the overall Oracle Database disk space requirements, see "Planning the Physical Architecture." For more information about sizing the core schema, see the discussion on managing the OSM database schema in the *OSM System Administrator's Guide*.

Tablespaces

The OSM installer prompts connect to a database using a user account with the sysdba privilege. You then select the following permanent database tablespaces:

- The default tablespace for all OSM schemas.
- Model Data and Model Indexes tablespaces: Used mainly for cartridge metadata and configuration data.
- Order Data and Order Indexes: Used for order data and auxiliary order-related tables.

For production instances, a minimum of two tablespaces should be created; one permanent and one temporary. For performance reasons and to facilitate backup and recovery, you should not share the permanent tablespaces for OSM with other applications. You should put model data and indexes on different tablespaces than order data and indexes.

For a production environment, you must partition your schema when running the OSM installer. If you choose the same tablespace for order data and order indexes, the OSM installer creates local index partitions with tablespace DEFAULT, which means that local index partitions are stored in the same tablespace as table partitions.

You can also create new table partitions in different tablespaces for increased administration and availability, for example on a rotation basis. If a tablespace is damaged, the impact and restoration effort could be limited to one or just a few partitions. See the discussion in the *OSM System Administrator's Guide* on adding partitions online or offline for more information.

Oracle recommends the following:

- Create tablespaces dedicated to OSM, so that OSM performance and availability are not affected by other applications, for example due to I/O contention or if a tablespace must be taken offline. Store the datafiles of these tablespaces on different disk drives to reduce I/O contention with other applications.
- Create locally managed tablespaces with automatic segment space management by specifying EXTENT MANAGEMENT LOCAL and SEGMENT SPACE MANAGEMENT AUTO in the CREATE TABLESPACE statement. Both options are the default for permanent tablespaces because they enhance performance and manageability.
- Configure automatic database extent management by using the AUTOALLOCATE clause of the CREATE TABLESPACE statement. This is the default. You cannot use uniform extents in the OSM database because the OSM installation will fail.

- If you use smallfile tablespaces, do not create hundreds of small datafiles. These files must be checkpointed, resulting in unnecessary processing. Note that Oracle Database places a limit on the number of blocks per datafile depending on the platform. The typical limit is 222-1, which limits the datafile size to 32 GB for 8K blocks.

Additional considerations if you use bigfile tablespaces:

- If data is stored in bigfile tablespaces instead of traditional tablespaces, the performance of database opens, checkpoints, and DBWR processes should improve. However, increasing the datafile size might increase time to restore a corrupted file or create a new datafile. You can mitigate the risk of corruption by using multiple tablespaces for partitions, for example on a rotating basis.
- Bigfile tablespaces are intended to be used with Automatic Storage Management (Oracle ASM) or other logical volume managers that supports striping or RAID, and dynamically extensible logical volumes.
- Avoid creating bigfile tablespaces on a system that does not support striping because of negative implications for parallel query execution and RMAN backup parallelization.
- Using bigfile tablespaces on platforms that do not support large file sizes is not recommended and can limit tablespace capacity.

For more information about managing tablespaces, see *Oracle Database Administration Guide*.

OSM data is placed in the permanent tablespace(s) and the temporary tablespace is used by the Oracle database as a workspace while processing OSM commands. The OSM data can be placed in one tablespace for a minimum installation, but OSM performs better when data is distributed across multiple tablespaces. You can use up to five tablespaces when initially installing the system.

It is possible to spread the OSM database over more than five tablespaces by altering the database installation and upgrade scripts. This must only be completed by an experienced Oracle DBA. For more information, contact Oracle.

In a high throughput system, each tablespace should be created on a different physical disk. This limits disk contention and IO bottlenecks to improve performance. It is recommended that the Oracle redo log files be placed on a separate, dedicated physical disk. You should not have any other load on this disk.

In a production system, a RAID device should be used for physical storage. In this case, there is no advantage to placing tablespaces on different physical RAID drives as long as space is available.

You can create tablespaces in either a traditional database instance or in a pluggable database instance.

The following is a bigfile tablespace creation script sample for a small installation model on an ASM diskgroup called +DATA.

```
create bigfile tablespace model_data
datafile '+DATA' size 100M;

create bigfile tablespace model_index
datafile '+DATA' size 100M;

create bigfile tablespace order_data
datafile '+DATA' size 200G;

create bigfile tablespace order_index
datafile '+DATA' size 200G;
```

> **Note**
>
> If you are using Chinese UTF8 characters, the Block Size for the tablespaces used by the OSM database must be configured for 8K at database instance creation.

5

Installing and Configuring the WebLogic Server Cluster

This chapter describes how you can install and configure a WebLogic Server cluster in preparation for an Oracle Communications Order and Service Management (OSM) installation.

Preparing WebLogic Server for an OSM Cluster Installation

You prepare a WebLogic Server clustered environment for an OSM installation by doing the following:

1. Prepare the operating system that will host WebLogic Server. See "Preparing the Operating System."
2. Install WebLogic Server. See "Installing WebLogic Server Software."
3. Create required database schemas using the Repository Creation Utility (RCU). See "Creating Database Schemas Using RCU."
4. Create the WebLogic Server domain and configure the required server instances and cluster. See "Creating the WebLogic Server Domain."
5. Replicate the WebLogic Server domain on all the machines within the domain. See "Replicating the Domain on Other Machines."
6. Configure the WebLogic Server domain and managed servers. See "Configuring the Domain and Managed Servers."

Preparing the Operating System

Use the following system and user limits:

- **Core file size**: Limit core file size to zero. If a core dump occurs or the JVM crashes, very large memory and data heaps might be written to the disk. Oracle recommends setting a positive value for core file size only if a crash occurs and must be debugged on the next occurrence.
- **Number of open files**: OSM typically references and loads large numbers of internal and third-party JAR files. Also, each application opens and maintains several configuration files, log files, and numerous network socket and JDBC connections. All these activities use a large number of open files, during startup and during application deployment and redeployment. Oracle recommends increasing the number of open file limits for OSM.
- **Number of user processes**: For the same reasons as increasing the number of open files, Oracle recommends increasing the limit for user processes.
- **Socket buffers**: To help minimize packet loss for the Coherence cluster, Oracle recommends setting the socket buffers of the operating system to at least 2 MB. For deployments with a high order processing rate, Oracle recommends 16 MB.
- **Number of clients**: Set `somaxconn` to at least **1024** to allow for a large number of client server connections.

- **Number of queued packets**: Set `netdev_max_backlog` to at least **32768** to minimize packet loss.

Table 5-1 summarizes the recommendations for configuring the operating system.

Table 5-1 Configuration Recommendations by Operating System

Configuration Item	Linux
core file size (soft)	0
core file size (hard)	0
open files (soft)	65536
open files (hard)	65536
max_user_processes (soft)	774889
max_user_processes (hard)	774889

> **Note**
>
> With engineered systems, such as Sparc SuperCluster or Exadata, most of the operating system tuning is preconfigured with appropriate values by default.

Installing WebLogic Server Software

The software for WebLogic Server and Application Development Framework (ADF) is included in the OSM software media pack. You download the OSM software media pack from the Oracle software delivery website:

`https://edelivery.oracle.com/`

You install WebLogic Server on all machines that will participate in your domain. The installation directories must be the same on all machines. For complete installation instructions and general information about installing and configuring WebLogic Server, see the WebLogic Server documentation.

> **Note**
>
> See *OSM Compatibility Matrix* for WebLogic Server version and patch information. Ensure that you use the WebLogic Server documentation specific to the required WebLogic Server version.

To install WebLogic Server:

1. Ensure that you have installed the 64-bit version of Java and the Java Development Kit (JDK) that is supported by OSM.
2. Set environment variables for the version of Java that is supported by OSM, not the version included with WebLogic Server. Do the following:
   - Set JAVA_HOME to the location of the supported Java version.
   - On a UNIX system, add **$JAVA_HOME/bin** to the PATH variable.

3. Install the WebLogic Server software as described in *Oracle Fusion Middleware Installing and Configuring Oracle WebLogic Server and Coherence*. When prompted for the installation type, select **Complete**.
4. Download and install any necessary patches from Oracle support. Follow the instructions in the **README.txt** file that is included with the patch.

Creating Database Schemas Using RCU

After you install the WebLogic Server software, create schemas in the database. You create the schemas using the Repository Creation Utility (RCU), which is included in the WebLogic Server installation. For complete information about using RCU, see *Oracle Fusion Middleware Creating Schemas with the Repository Creation Utility*.

The schemas are required for creating the WebLogic Server domain. Each schema can be used by only one domain. If you create a new domain, you must also create new schemas.

Before creating the schemas, ensure that you have your database connection string, port, administrator credentials, and service name ready.

When you create database schemas using RCU, keep the following considerations in mind:

- When entering database details for an Oracle RAC instance, use the host name of one of the Oracle RAC instances. Do not use the SCAN IP address.
- When selecting components, ensure that you create a new prefix and select all of the following components:
  - Common Infrastructure Services (*prefix*_STB)
  - Oracle Platform Security Services (*prefix*_OPSS)
  - Audit Services (*prefix*_IAU)
  - Audit Services Append (*prefix*_IAU_APPEND)
  - Audit Services Viewer (*prefix*_IAU_VIEWER)
- When prompted for passwords, select **Use the same password for all schemas** and enter the password of your choice.
- Use the default tablespace configuration provided by the installer.

> **Note**
>
> RCU may take several minutes to create the schemas. If creating the schemas takes an unusual amount of time, Oracle recommends that you purge the database recycle bin to ensure that RCU schemas are created more quickly the next time. For more information, see "OSM and RCU Installers Are Slow to Run Database Tablespace Query."

Creating the WebLogic Server Domain

This section describes how to create the WebLogic server domain and configure the required server instances and cluster using the Fusion Middleware Configuration Wizard.

For more information about WebLogic clustering, refer to the WebLogic Server documentation. For more information about the Configuration Wizard, including detailed descriptions of the

Configuration Wizard screens, see *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*.

To configure the WebLogic server domain for an OSM cluster:

1. Ensure the following:
   - If you are configuring the managed servers for whole server migration, verify that you have available interfaces on each machine hosting a managed server. For example, eth0. Ensure that the IP address you want to use for each managed server is configured with a host name on a DNS server.

     Note

     After you have completed all procedure relating to whole server migration, node manager automatically creates floating IP addresses for each managed server when you start managed servers from the Remote console. For example, assuming you have configured a machine to run two managed servers, node manager would create eth0:1 for managed server 1 and eth0:2 for managed server 2.

   - If you are configuring managed servers without whole server migration, ensure that you have static IP addresses and port numbers for each managed server.

     Tip

     Oracle recommends that you use DNS host names to facilitate any IP address changes during the lifetime of the domain.
2. Ensure that the WebLogic Server software has been installed on each machine that will be part of the domain.
3. Log in to the machine that will be hosting the administration server in your domain.
4. On UNIX platforms, run the following command:

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

   The Configuration Wizard launches and the **Configuration Type** screen is displayed.
5. Ensure that **Create a new domain** is selected, and enter the path to the domain in the **Domain Location** field, and click **Next**.

   The **Templates** screen is displayed.
6. Ensure that **Create Domain Using Product Templates** is selected, choose the following templates from the Available Templates list:

   On the **Templates** screen:
   - Select the **Oracle JRF** and **WebLogic Coherence Cluster Extension** templates.
   - Select the **Oracle Enterprise Manager** template if you want to use Oracle Enterprise Manager Fusion Middleware Control to view and manage OSM logs.
   - The **Basic WebLogic Server Domain** template is selected by default and you cannot deselect it.
7. Click **Next**.
8. Do one of the following:

- If you selected **Oracle Enterprise Manager**, the **Application Location** screen is displayed. Go to step 9.
- If you did not select **Oracle Enterprise Manager**, the **Administrator Account** screen is displayed. Go to step 11.

9. In the **Application location** field, enter the path to the application directory.
10. Click **Next**.

    The **Administrator Account** screen is displayed.
11. In the **Name** field, enter the WebLogic Administrator user name.
12. In the **Password** field and again in the **Confirm password** field, enter the WebLogic Administrator password.

    > **Note**
    >
    > Alphanumeric characters are mandatory in the Password field.

13. Click **Next**.

    The **Domain Mode and JDK** screen is displayed.
14. Select **Production Mode**. Selecting this ensures that the default optimization settings are applied. Disable Secure Production Mode by selecting the checkbox **Disable Secure Mode**.
15. Select the default JDK, or browse to the location of a 64-bit JDK, and click **Next.**

    The **Database Configuration Type** screen is displayed.
16. Connect to the database by doing the following:
    a. Select the **RCU Data** option.
    b. In the **Vendor** and **Driver** drop-down lists, use the default selection.
    c. In the **DBMS/Service** field, enter the database DBMS name, or service name if you selected a service type driver.
    d. In the **Host Name** field, enter the name of the server hosting the database. The host name is the Oracle RAC instance you used to create the RCU repository.
    e. In the **Port** field, enter the port number on which the Oracle RAC instance listens.
    f. In the **Schema Owner** field, enter the user name for connecting to the service table schema. For example:

       *prefix*_**STB**

       where *prefix* is the new prefix you created on the **Select Components** screen of RCU.
    g. In the **Schema Password** field, enter the password that you created for the schemas on the bscreen of RCU.
    h. Click **Get RCU Configuration**. The Configuration Wizard tests the connection to the database and displays the results in the Connection Result Log area.
    i. Click **Next**.

    The **JDBC Component Schema** screen is displayed with pre-loaded configuration data for the RCU schemas.

17. Verify the pre-loaded schema configuration data for each of the following schemas in the component schema table:
    - OPSS Audit Schema
    - OPSS Audit Viewer Schema
    - OPSS Schema
    - LocalSvcTbl Schema
18. Select all schemas in the component schema table.
19. Select **Convert to RAC multi data source**.
20. Click **Next**.

    The **Oracle RAC Multi Data Source Component Schema** screen is displayed.
21. In the first row of the **Host Name** column, enter the host name of the second Oracle RAC instance.
22. In the first row of the **Instance Name** column, enter the SID of the second Oracle RAC instance.
23. In the first row of the **Port** column, enter the port number of the Oracle RAC listener.
24. If you have additional Oracle RAC instances, click **Add Hosts** to add new Oracle RAC instance rows and repeat steps 21 to 23.
25. In the **Service Name** field, enter the service name.
26. Click **Next**.

    The **JDBC Component Schema Test** screen is displayed.
27. Select all of the component schemas in the table and click **Test Selected Connections**.

    If any of the tests fail, go back to the previous screens and check that you entered the correct values in the fields.

    When all of the tests are successful, click **Next**.

    The **Advanced Configuration** screen is displayed.
28. Select **Administration Server**, **Node Manager, Managed Servers, Clusters and Coherence**, and **Deployments and Services**, and click **Next.**

    The **Administration Server** screen is displayed.
29. In the **Server Name** field, enter a name for the administration server.
30. In the **Listen Address** field, enter a value for the listen address. For example, an IP address or DNS name. Oracle recommends that you use a specific host name in the Listen Address field to facilitate any future IP address changes that may occur. In addition, avoid selecting All Local Addresses because this causes WebLogic Server to bind to every available IP address on the machine, potentially reducing performance.
31. In the **Listen Port** field, enter a value for the listen port.
32. (Optional) Select **Enable SSL** and enter a value for the SSL listen port in the **SSL Listen Port** field. Oracle recommends enabling SSL and configuring an SSL listen port to ensure secure communication over the Internet.
33. Click **Next**.

    The **Node Manager** screen is displayed.
34. Select **Per Domain Default Location** for the Node Manager type.
35. In the **Username** field, enter a user name for Node Manager.

36. In the **Password** field and again in the **Confirm Password** field, enter a password for Node Manager.

    The Node Manager uses the user name and password that you provide to authenticate connections between Node Manager and clients. They are independent from the server administration credentials.

37. Click **Next**.

    The **Managed Servers** screen is displayed.

38. Click **Add**, which creates a managed server. Repeat this step for any additional managed servers.

    > **Note**
    >
    > Oracle highly recommends that you set up a dedicated external load balancer, even for your development systems.

39. Enter managed server names, IP addresses or host names (Oracle recommends DNS host names for high availability), and port numbers. If required, enable SSL and enter an SSL listen port. Click **Next**.

    The **Clusters** screen is displayed.

40. Click **Add**, which adds a cluster row.

41. In the **Name** field, enter a name.

42. In the **Cluster address** field, add the address for the cluster. When you are using an external load balancer, this needs to be the address of the DNS name that maps to the IP addresses or DNS names of each WebLogic managed server in the cluster. The managed servers need to have addresses with different IP addresses or DNS names but with the same port numbers. For information on how to set this up, refer to the documentation for your load balancer.

43. Click **Next**.

    The **Assign Servers to Clusters** screen is displayed.

44. In the Server area, select all managed servers.

45. Click the right arrow, which moves all managed servers under the cluster.

46. Click **Next**.

    The **HTTP Proxy Applications** screen is displayed. You do not need to make any changes here.

47. Click **Next**.

    The **Coherence Clusters** window is displayed.

48. In the **Name** field, enter a name for the Coherence cluster.

49. In the **Unicast Listen Port** field, leave the default value of **0**.

50. Click **Next**.

    The **Machines** screen is displayed.

51. On the **Unix Machine** tab:

    a. Click **Add** to add any UNIX machines in your domain that will run Node Manager.

b. Enter machine names, Node Manager listen addresses, and Node Manager listen ports.

c. (Optional) Select **Post bind GID enabled** and enter the UNIX group ID under which a server on this machine will run after it finishes all privileged startup actions.

d. (Optional) Select **Post bind UID enabled** and enter the UNIX user ID under which a server on this machine will run after it finishes all privileged startup actions.

52. Click **Next**.

    The **Assign Servers to Machines** window is displayed.

53. Add the available servers to the appropriate machines as decided upon in your configuration details. Click **Next.**

    The **Deployments Targeting** screen is displayed.

54. Verify that all of the deployments under the **Library** and **Application** folders are targeted to your new cluster and the administration server. If they are not, assign deployments to your cluster and administration server as follows:

    a. In the **Targets** list, select the cluster where you want to install OSM.

    b. In the **Deployments** list, select the **Library** and **Application** folders.

    > **Note**
    >
    > Do not select any deployments to be assigned to the proxy server (if used).

    c. Click the right arrow to include the OSM components you want to install on the cluster.

55. Click **Next**.

    The **Services Targeting** screen is displayed.

56. Target services to your cluster:

    a. In the **Targets** list, select the cluster where you want to install OSM.

    b. In the **Services** list, select all of the services that you want install on the cluster.

    c. Click the right arrow to assign the services to the cluster.

    d. Click **Next**.

    The **Configuration Summary** screen is displayed, which provides a summary of the applications, services, and libraries to be deployed in the domain.

57. Review the information in the **Configuration Summary** screen and confirm that the cluster organization matches your requirements. If you find any discrepancies, click the **Previous** button to return to the appropriate screen and make the necessary changes. When you are done, click **Create**.

    The **Configuration Progress** screen shows the progress of your domain creation.

58. When the domain creation process completes and the **Configuration Success** screen is displayed, click **Finish**.

Replicating the Domain on Other Machines

The newly created domain is now installed on a single machine. This section describes the steps necessary to replicate the domain installation on other machines. WebLogic provides two utilities to do this: pack and unpack.

> **Note**
>
> If the `unpack.jar` command fails with a write error, see "Command for unpack.jar Fails with a Write Error" for troubleshooting information.

Starting and Configuring Credentials on the First Machine

Before you can replicate the domain to another machine, you must first start nodemanager and the administration server on the first machine. You can then create a **boot.properties** file for the administration server to enable each server to start without prompting you for the administrator username and password.

To create the **boot.properties** file:

1. Open a terminal.
2. Go to *domain_home*/**bin** for your base domain.
3. Run the following command which starts nodemanager:

   ```
   startNodeManager.sh
   ```
4. Open a second terminal.
5. Go to *domain_home*/**bin** for your base domain.
6. Run the following command which starts the administration server:

   ```
   startWebLogic.sh
   ```
7. When the following text appears, enter the administration server username:

   ```
   Enter username to boot WebLogic server:
   ```
8. When the following test appears, enter the administration server password:

   ```
   Enter password to boot WebLogic server:
   ```
9. After the administration server is in the running state, open a third terminal.
10. Go to the *domain_home*/**servers**/*AdminServer* directory (where *AdminServer* is the name of the administration server.)
11. Create the following directory:

    ```
    mkdir -p security
    ```
12. In the directory you just created, create the **boot.properties** file.

    ```
    touch boot.properties
    ```
13. Using a text editor, add the following lines to the file:

    username=*username*
    password=*password*

    where

    - *username* is the administrator user name for the WebLogic administration server.
    - *password* is the password for the WebLogic administration server.

    These values are entered in clear text but will be encrypted when you start the server for the first time.
14. Save and close the **boot.properties** file.

15. Close the third terminal.
16. In the second terminal, stop the administration server using **Ctrl-C**.
17. Close the second terminal.
18. In the first terminal, stop node manager using **Ctrl-C**.
19. Close the first terminal.

Creating a Domain Template for Use on Other Machines

To create a domain directory (template) that can be used on other machines within the domain:

1. On the machine that contains the administration server and the definition of managed servers, go to the *FMW_HOME***/oracle_common/common/bin** directory.
2. Run the following command:

   pack.sh -domain=*domain_home* **-template=***template***.jar -template_name="***template_name***"**

   where:

   - *domain_home* is the full or relative path of the WebLogic domain from which the template is to be created
   - *template* is the full or relative path of the template, and the filename of the template to be created
   - *template_name* is a descriptive name for the template

   For example:

   ```
   pack.sh -domain=/opt/oracle/Middleware/user_projects/domains/cluster_demo -
   template=/opt/oracle/Middleware/user_projects/domains/cluster_demo.jar -
   template_name="cluster_demo"
   ```

Replicating the Domain Template on Other Machines

Use the following steps to replicate the created template file to all other machines in the domain.

1. Establish a session with the remote machine and copy the template to it.
2. Navigate to the *FMW_HOME***/oracle_common/common/bin** directory.
3. Run the following command:

   unpack.sh -template=*template***.jar -domain=***domain*

   where:

   - *template* is the full or relative path of the template that you copied to the remote machine
   - *domain* is the full or relative path of the domain to be created

   For example:

   ```
   unpack.sh -template=/opt/oracle/Middleware/user_projects/domains/cluster_demo.jar -
   domain=/opt/oracle/Middleware/user_projects/domains/cluster_demo
   ```

Starting the Administration Server

To start the WebLogic Server administration server:

1. Navigate to the *domain_home* directory of the machine that runs the administration server, and start the server by running the following command:

   ```
   nohup ./startWebLogic.sh 2>&1 &
   ```

2. Verify that the administration server starts properly by running the following command:

   ```
   tail -f nohup.out
   ```

   A log entry should indicate that the server is running with the following line:

   ```
   Server started in RUNNING mode
   ```

Configuring the Domain and Managed Servers

The following sections describe how you should configure the domain and managed servers.

Configuring Oracle Coherence for an OSM Cluster

This section provides configuration suggestions and best practices to avoid conflicts with Oracle Coherence in a clustered OSM environment.

For information about configuring and troubleshooting Oracle Coherence, refer to the Coherence documentation. For performance tuning details, see the Coherence Knowledge Base website:

https://docs.oracle.com/en/middleware/fusion-middleware/coherence/14.1.2/administer/performance-tuning.html#COHAG217

Increasing Buffer Sizes to Support Coherence

Oracle recommends that you configure your system for larger buffers:

On Oracle Linux and Red Hat Enterprise Linux, run the following commands as root:

```
sysctl -w net.core.rmem_max=2097152
sysctl -w net.core.wmem_max=2097152
```

> **Note**
>
> This change can be made permanent by adding, or changing the values of, the following parameters in **/etc/sysctl.conf**:
>
> ```
> net.core.rmem_max=2097152
> net.core.wmem_max=2097152
> ```

Preventing Unnecessary Use of Swap Space

Do not use swap space in an OSM production environment for the WebLogic Servers if at all possible because it can significantly increase the duration of full garbage collection. Setting the

swappiness to 10 ensures that the operating system only uses swap space when absolutely necessary.

To set swappiness to 10, do the following:

1. On a machine running a managed server, log into a terminal as root user.
2. Run the following command:

```
echo vm.swappiness=10 >> /etc/sysctl.conf
```

3. Repeat this procedure for all other machines running managed servers.

Securing Coherence

When you create a WebLogic Server with a cluster, the OSM installer automatically configures the Oracle Coherence connections and well known addresses (WKA) for the cluster, but does not automatically configure security settings for the cluster. For example, the installer does not configure authorized hosts.

Oracle recommends securing the coherence connections. For more information, see *Oracle Fusion Middleware Securing Oracle Coherence*.

Configuring Coherence for Load Balancing

After upgrading to this OSM release or on a new installation of this release, when running WebLogic 14.1.2 , the Coherence cache is loaded on only one node in a two-node cluster.

Coherence does not load balance the cache when there are only two nodes in the cluster. Because all the cache entries are in one node, all the requests are processed by that node only. This is an expected behavior in Coherence 12.2.1.2.x, 12.2.1.3.x, 12.2.1.4 and 14.1.2.

To avoid this behavior, if you use a two-node cluster, add the following command line option to the startup parameters:

```
-Dcoherence.distribution.2server=false
```

This option configures Coherence to load balance and distribute cache data on both the nodes in a two-node cluster setup.

Configuring Maximum Message Size

To configure the maximum number of bytes allowed in messages that are received over all supported protocols, add the following command line option to the startup parameters:

```
-Dweblogic.MaxMessageSize=300000000
```

The default is 10000000 bytes and this can cause errors (for example, BEA-000403 error message) when this limit is reached. For details about the error, see Diagnosing BEA-000403 Error Message (Doc ID: 1493101.1) on My Oracle Support.

Configuring Node Manager on All Machines in the Domain

This section describes how to configure the machines in your domain that will host Node Manager. OSM recommends using Node Manager to automatically restart managed servers after unexpected failure. Oracle also recommends that you configure whole server migration

Configuring Node Manager for Starting and Stopping Managed Servers

For each machine that will host Node Manager, do the following:

1. Open the *domain_home*/**nodemanager/nodemanager.properties** file.
2. Set the following values:

```
StartScriptEnabled=true
StopScriptEnabled=true
```

Configuring Node Manager for Whole Server Migration

You can optionally use node manager to create floating IP addresses for the managed servers in the cluster. This enables a managed server to migrate from one machine to another in the event of repeated server recover failure.

For each machine that will host Node Manager with whole server migration enabled, do the following:

1. Open the *domain_home*/**nodemanager/nodemanager.properties** file.
2. Set the following values:

```
CrashRecoveryEnabled=true
ethx=ip_address_range,NetMask=networkmask
UseMACBroadCast=true
```

where:

- *x* is the interface number that the floating IPs are created on. For example, eth0.

  When node manager creates the floating IP address, it adds **eth***x***:***n* where *n* is the number it assigns to a floating IP. For example, if a machine were to host two managed servers using the eth0 interface, node manager would create the eth0:1 and eth0:2 floating IP addresses. If a third managed server were to migrate to the machine, it would receive eth0:3. Node manager automatically creates and removes these interfaces as required.
- *ip_address_range* is the range of IP addresses associated to managed servers on the cluster that node manager can use as floating IP addresses. For example, you can specify a single IP address for a managed server, or a range of IP addresses associated to managed servers, such as 192.168.56.201-192.168.56.204, or * to specify all IP addresses of every managed server in the cluster.
- *networkmask* is the net mask for the interface for the floating IP. *networkmask* should be the same as the net mask on the interface. This parameter is optional.

Configured Whole Server Migration Floating IP Controls

For each machine with a node manager configured with whole server migration, do the following:

1. Set the UNIX PATH environment variable to the directories that contain the WebLogic Server files indicated in Table 5-2.

Table 5-2 Files Required for the PATH Environment Variable

File or Directory	Located in This Directory	Example (bash)
wlsifconfig.sh	*domain_home*/bin/server_migration	export WLSIFCONFIG_HOME=$DOMAIN_HOME/bin/server_migration export PATH=$PATH:$WLSIFCONFIG_HOME
wlscontrol.sh	*domain_home*/bin/nodemanager	export WLSCONTROL_HOME=$DOMAIN_HOME/bin/nodemanager export PATH=$PATH:$WLSCONTROL_HOME
node manager	*domain_home*/nodemanager	export NODEMANAGER_DOMAINS_HOME=$DOMAIN_HOME/nodemanager export PATH=$PATH:$NODEMANAGER_DOMAINS_HOME

2. Grant sudo privileges to the WebLogic Server user account so that the **wlsifconfig.sh** script can:
   - Work without a password prompt.
   - Have execute privilege on the **/sbin/ifconfig** and **/sbin/arping** binaries that are involved in creating floating IP address.

   To grant sudo privileges, do the following:

   a. Log in as the **root** user.

   b. Enter the following command:

   ```
   /usr/sbin/visudo
   ```

   The **/etc/sudoers** file opens in a text editor.

   c. Add the following line:

   ```
   Defaults:weblogic_user !requiretty
   weblogic_user ALL=NOPASSWD: /sbin/ifconfig,/sbin/arping
   ```

   where *weblogic_user* is the user account you created for running WebLogic Server.

3. Run the following commands to test whether the wlsifconfig.sh script is functioning:

   a. List all IP addresses configured on the interface.

   ```
   export ServerDir=/tmp
   wlsifconfig.sh -listif ethx
   ```

   where *x* is the interface number. For example

   ```
   export ServerDir=/tmp
   wlsifconfig.sh -listif eth0
   eth0 192.168.56.26
   ```

   b. Assign a floating IP address of one of the managed servers you want to node manager to manage.

   ```
   wlsifconfig.sh -addif ethx ip_address netmask
   ```

   where *ip_address* and *netmask* are the floating IP address and network mask you want to configure. For example:

   ```
   wlsifconfig.sh -addif eth0 192.168.56.124 255.255.255.0
   Generated command - sudo /sbin/ifconfig eth0:1 192.168.56.124 netmask
   255.255.255.0
   Successfully brought 192.168.56.124 netmask 255.255.255.0 online on eth0:1
   ```

   c. Confirm that the floating IP address was added.

```
/sbin/ifconfig
...
eth0:1  Link encap:Ethernet  HWaddr 08:00:27:5A:C0:09
        inet addr:192.168.56.124  Bcast:192.168.56.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
...
```

d. Remove the floating IP address.

```
wlsifconfig.sh -removeif ethx ip_address
```

For example:

```
wlsifconfig.sh -removeif eth0 192.168.56.124
Successfully removed 192.168.56.124 netmask  from eth0:1.
```

4. If any command fails, verify that the **/etc/sudoers** file is properly configured.

Enrolling Each Machine with the Domain

Each machine that will host Node Manager to start and stop the managed servers must be enrolled with the domain. This procedure is applicable to all node manager configurations.

To enroll each machine that will host Node Manager:

1. Start the administration server.
2. Run *WebLogic_home***/common/bin/wlst.sh** tool.
3. At the command prompts, run the following commands:

```
connect('username', 'password', 't3://ip_address:port')

nmEnroll('domain_home', 'domain_home/nodemanager')

exit()
```

where

- *username* is the administrator user name for the WebLogic administration server.
- *password* is the password for the WebLogic administration server.
- *ip_address* is the IP address for the WebLogic administration server.
- *port* is the port number for the WebLogic administration server.

Starting Node Manager on Each Machine

After Node Manager has been prepared on each machine, start them. For each machine, follow these steps:

1. Navigate to *WebLogic_home***/server/bin/**.
2. Start Node Manager by running the following command:

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

3. Verify that the Node Manager starts properly by running the following command:

```
tail -f nohup.out
```

A log entry should indicate that Node Manager is running with the following line:

```
INFO: Secure socket listener started on port port
```

where *port* is the port number used by Node Manager.

Configuring a Multicast IP Address for the Cluster Messaging Mode

By default, the WebLogic Configuration Wizard enables unicast messaging mode. For greater reliability and scalability, switch to multicast cluster messaging mode. See "About the WebLogic Messaging Mode and OSM Cluster Size" for more information about which messaging mode option to choose.

To configure a multicast IP address for the cluster messaging mode:

1. Configure the multicast IP address on your machine. For example, on an Oracle Linux machine, you can do the following:
   a. From the desktop, select **System**, then **Preferences**, then **Network Connections**.

      The Network Connections screen appears.
   b. Select the interface you want to configure a multicast IP address on, and click **Edit**.

      The Editing System screen appears.
   c. Click the **IPv4 Settings** tab.
   d. Click **Routes…**

      The Editing IPv4 routes for System screen appears.
   e. In the **Address** field, enter the multicast IP address you want to use for cluster communications.
   f. In the **Netmask** field, enter a network mask.
   g. Click **OK**.
   h. Click **Apply**.
   i. Click **Close**.
   j. In the top right hand corner, click the Network icon.
   k. Below the interface you have configured an multicast IP address for, click **Disconnect**.
   l. When the interface has disconnected, click **Connect**.
   m. Open a terminal.
   n. Go to *WebLogic_home***/server/bin**.
   o. Source the setWLSEnv.sh script.

      ```
      source ./setWLSEnv.sh
      ```
   p. Run the following command from the machine:

      ```
      java utils.MulticastTest -N machine_name -A multicast_ip -P multicast_port
      ```

      where

      - *machine_name* is any name that identifies the sender of the message. Use a different name for every test you start.
      - *multicast_ip* is the multicast IP address you want to use for the cluster. Use any IP address between 224.0.0.0 and 239.255.255.255. If there are any running WebLogic clusters using multicast, ensure that you do not use the same IP address during this test.

- *multicast_port* is the multicast port number you want to use for the cluster. If there are any other applications running on the machine that use multicast, ensure that you do not use the same port.

For example:

```
java utils.MulticastTest -N osm_test_from_machine1 -A 239.192.0.0 -P 7521
***** WARNING ***** WARNING ***** WARNING *****
Do NOT use the same multicast address as a running WLS cluster.


Starting test.  Hit any key to abort


Using multicast address 239.192.0.0:7521
Will send messages under the name osm_test_from_machine1 every 2 seconds
Will print warning every 600 seconds if no messages are received

        I (osm_test_from_machine1) sent message num 1
New Neighbor sm_test_from_machine1 found on message number 1
Received message 2 from sm_test_from_machine1
        I (osm_test_from_machine1) sent message num 2
Received message 3 from sm_test_from_machine1
        I (osm_test_from_machine1) sent message num 3
etc...
```

2. Start the administration server as described in "Starting the Administration Server."
3. Log in to the WebLogic Remote Console.
4. Click **Edit Tree**.
5. Expand **Environment**, and select **Clusters**.
6. From the table of clusters, select the OSM cluster.
7. Select the **Messaging** tab.
8. From the **Cluster Messaging Mode** drop-down list, select **Multicast**.
9. Click the **Show Advanced Fields** checkbox.
10. In the **Multicast address** field, enter a multicast address. The multicast address range can be between 224.0.0.1 to 239.255.255.255.
11. In the **Multicast port** field, enter a multicast port number. The multicast port can be between 1 to 65535.
12. In the **Idle Periods Until Timeout** field, enter 5.
13. Click **Save**.
14. Select the **Health** tab.
15. In the **Heath Check Interval** field, enter 20000.
16. Click **Save**.
17. Click the shopping cart. From the shopping cart, select **Commit changes**.

Preventing Connection Timeout when Using a Remote Database

To prevent a connection timeout when your database and server are in separate locations, do the following:

1. Increase the value of the Stuck Thread Max time parameter as follows:

a. Log in to the WebLogic Remote Console.

b. From the **Edit Tree**, expand **Environment**, and then select **Servers**.

   The **Summary of Servers** page is displayed.

c. Click the name of the WebLogic server where you want to deploy the cartridges.

   The configuration parameters for the server are displayed on a tabbed page.

d. Select **Advanced**. From **Advanced**, select the **Tuning** tab and modify the value of the **Stuck Thread Max Time** parameter to an appropriate value above 1200 seconds. WebLogic considers a thread to be stuck when the thread takes more than a specified amount of time to process a single request. Oracle recommends that you set this value to an optimal level based on performance and stress testing.

e. Click **Save**.

2. Increase the value of the Timeout Seconds Java Transaction API parameter (JTA timeout) as follows:

   a. In the **Edit** tree of the WebLogic Remote Console, click **Services**.

      The configuration parameters for the domain are displayed on a tabbed page.

   b. Click the **JTA** and modify the value of the **Timeout Seconds** parameter to an appropriate value. In most cases, a value of 600 seconds is enough.

      > **Note**
      >
      > If the value is less than 600 when you run the Installer, you are prompted to increase it during installation.

   c. Click the **Security** tab and select **Anonymous Admin Lookup Enabled**.

   d. Click **Save**.

   > **Note**
   >
   > The Java Transaction API parameter applies to the domain globally. After you have installed OSM, you can configure an oms-config.xml parameter that controls this value dynamically when OSM deploys or un-deploys a cartridge. For more information, see "Preventing Connection Timeout Issues During Cartridge Deployment".

See the WebLogic Server documentation for more information about these parameters.

Recommended Configuration for WebLogic Servers for Production Systems

Recommendations for creating and configuring managed servers for OSM include the following:

- **64-bit flags**: Ensure to use the **-d64** option for each server so that available native libraries or performance packs are used when you start WebLogic servers.
- **Logging**:

  - Consider limiting the number of log files, the log file maximum size, and rotation policy. Limit the number of log files to 10 and ensure log files are rotated at start up so that a fresh log file is available.
  - Consider turning off logging messages to the domain for each managed server. WebLogic server instances send messages to a number of destinations, including the local server log file and the centralized domain log, which can affect performance. For more information, see the chapter about understanding WebLogic logging services in the document *Oracle Fusion Middleware Configuring Log Files and Filtering Log Messages for Oracle WebLogic Server*.

- **JDBC logging**: Disable JDBC logging in production systems because it has a substantial impact on performance.
- **WebLogic networking**: Enable native input/output (IO). WebLogic uses software modules called muxers (multiplexers) to read incoming requests on the server and incoming responses on the client. To maximize network socket performance, ensure to use native, platform-optimized muxers.
- **Transaction timeouts**: Set timeouts for correct rollback handling, keeping in mind that global transactions span multiple transaction sources, such as JMS and JDBC.
- **Longest transaction time**: Set the longest time a transaction can be active. The global transaction timeout, which is the longest time that a transaction can be active, is determined by the Java Transaction API (JTA) timeout. All transactions typically complete within only a few seconds (excluding the first transaction after server startup). You can determine the optimal level for your system based on performance and stress testing. Change the JTA timeout value by using the WebLogic Remote Console'.

> **Note**
>
> If you are using Oracle Exalogic Elastic Cloud, enable optimizations in order to improve thread count management and request processing, and to reduce lock contention.

Oracle recommends that you do not leave the WebLogic Server listen address undefined on a computer that uses multiple IP addresses. In this case, the server binds to all available IP addresses, which slows down server startup time. Bind a WebLogic server to a fully qualified host name, rather than an IP address. This ensures that SSL server-to-server communication works correctly without requiring host name verification. It also allows administrators to change IP addresses without reconfiguring WebLogic.

Table 5-3 includes a summary of the recommended WebLogic server configurations.

Table 5-3 Configuration Recommendations for the Server

Configuration Item	Value	Notes
JTA Timeout	600	Set this value for better performance.
Enable Exalogic Optimizations	true	When you are using an Exalogic server.
-d64	enabled	Ensure that you use 64-bit native libraries in the startup paths.
Log: Rotate log on startup	true	Not Applicable
Native IO enabled	true	Not Applicable
Log: Number of files limited	true	Not Applicable

Table 5-3 (Cont.) Configuration Recommendations for the Server

Configuration Item	Value	Notes
Log: File count	10	Not Applicable
All file-based persistent stores use synchronous write policy	Direct-Write-With-Cache	Not Applicable
All JMS Servers > Persistent Stores	enabled	File-based persistence has better performance that a JDBC store, but JDBC offers consistent backup snapshots of OSM data and JMS messages.

Configuring Managed Server Startup Parameters

Oracle recommends that you use the startup parameters specified in this section for each managed server in your cluster. These ensure properly configured managed servers and settings such as the Java heap, garbage collection logging, and so on. These startup parameters only work when using node manager.

To configure the managed server startup parameters:

1. Log in to the WebLogic Remote Console.
2. Click **Edit Tree**.
3. From the **Edit Tree**, expand **Environment**. From **Environment**, select **Servers**.

   The Summary of Servers screen is displayed.
4. For each managed server, select the server's name.

   Then select the **Advanced** tab.
5. Click the **Node Manager** subtab.
6. In the Arguments field, enter the following settings:

```
-server
-Djava.net.preferIPv4Stack=true
-Dtangosol.coherence.ipmonitor.pingtimeout=9s
-Dtangosol.coherence.log=domain_home/servers/managed_server/logs/
managed_server_coherence.log
-Dweblogic.jdbc.remoteEnabled=false
-Xms31g -Xmx31g
-Xmn14g
-XX:+UseCompressedOops
-XX:-UseAdaptiveSizePolicy
-XX:SurvivorRatio=3
-XX:TargetSurvivorRatio=70
-XX:CodeCacheMinimumFreeSpace=16m
-XX:ReservedCodeCacheSize=256m
-XX:+UseParallelOldGC
-XX:ParallelGCThreads=processors
-verbose:gc
-XX:+PrintGCDetails
-XX:+PrintGCDateStamps
-XX:+PrintGCTimeStamps
-XX:+PrintTenuringDistribution
-XX:+PrintAdaptiveSizePolicy
-Xloggc:domain_home/servers/managed_server/logs/managed_server_gc%t.log
-XX:+DisableExplicitGC
```

```
-XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=domain_home/servers/managed_server/logs/
-XX:+UnlockCommercialFeatures
-XX:+FlightRecorder
-XX:+ParallelRefProcEnabled
-XX:+AlwaysPreTouch
```

where:

- *processors* is the number of processors you want to allocate to the managed server. If you have more than one managed server running on the system, ensure that you have enough processors available for each.
- *managed_server* is the name of the managed server.

7. Click **Save**.
8. Click the shopping cart. From the shopping cart, select **Commit changes**.

Configuring Cluster Settings

Before you install OSM, you must prepare the domain by enabling the WebLogic plug-in, and configuring the hardware load balancer.

To prepare the domain for OSM installation:

1. Log in to the WebLogic Remote Console.
2. If you are using HTTPS for communicating with the OSM web clients, do the following:
   a. From the **Edit Tree**, expand **Environment** and then select **Servers**.
   b. Select either the administration server or one of the managed servers.
   c. From the **General**tab, select **Advanced**.
   d. Expand **Advanced**.
   e. Enable **WebLogic Plug-In Enabled**.
   f. Click **Save**.
   g. Repeat steps 2.b to 2.f for all managed servers and the administration server.
3. If you are running a hardware load balancer, do the following:
   a. From the **Edit Tree**, expand **Environment**. From **Environment**, select **Clusters**.
   b. Select the cluster to which you are installing OSM.
   c. Select the **HTTP** tab.
   d. In the **Frontend Host** field, enter the host name or IP address of the load balancer.
   e. In the **Frontend HTTP Port** field, enter the HTTP port of the load balancer.
   f. In the **Frontend HTTPS Port** field, enter the HTTPS port of the load balancer.
   g. Click **Save**.

Configuring Server Settings

To configure server settings, do the following:

1. Log in to the WebLogic Remote Console.
2. Click **Edit Tree**.

3. From the **Edit Tree**, expand **Environment**. From **Environment**, select **Servers**.

   The Summary of Servers screen is displayed.
4. Select a server (for example, the administration server, or a managed server).
5. Click the **Logging** tab.
6. In the **General** sub tab, click **Advanced**.
7. In the Domain log broadcaster area, from the **Severity Level** list, select OFF.
8. Click **Save**.
9. Repeat steps 3 to 8 for all other servers in the cluster.
10. From the **Edit tree**, expand **Environment** and select **Servers**.

    The Summary of Servers screen is displayed.
11. Select a server.
12. Click the **Services** tab.
13. From the **Default Store** sub tab, select the **Synchronous Write Policy** list. From the **Synchronous Write Policy**, select **Direct-Write-With-Cache**.
14. Click **Save**.
15. Repeat steps 10 to 14 for all other servers in the cluster.
16. From the Domain Structure tree, expand **Environment** and select **Servers**.

    The Summary of Servers screen is displayed.
17. Select a server.
18. In the **Logging** tab, click the **HTTP** sub tab.
19. Deselect **HTTP access log file enabled**.
20. Click **Save**.
21. Repeat steps 16 to 20 for all other servers in the cluster.
22. Click the shopping cart. From the shopping cart, select **Commit changes**.

Starting and Verifying all Machines in the Cluster

This section contains information about starting the WebLogic Server cluster and verifying the setup of the cluster.

To start and verify the cluster:

1. Go to *domain_home***/bin** for your base domain.
2. Run the following command which starts the administration server:

   ```
   startWebLogic.sh
   ```
3. Go to *domain_home***/bin** for your base domain on each machine running a managed server.
4. Run the following command:

   ```
   startupscript.sh server_name http://IP_address:port
   ```

   where:

   - *startupscript* is the name of the startup script for the managed server you are starting.

- *server_name* is the name of the managed server you are starting.
- *IP_address* is the IP address of the administration server.
- *port* is the port number of the administration server.

5. Log in to the WebLogic Server Remote Console.

   The Remote Console is displayed.

6. Click **Environment**. From **Environment**, select **Servers**.

   The summary page shows which servers belong to the cluster along with their configured listen address, port, state, and health status.

Configuring Whole Server Migration

The procedures in the section are applicable if you want to use whole server migration and have done the following:

- Configured floating IP addresses for the managed server. See "Creating the WebLogic Server Domain" for more information.
- Configured node manager properties for whole server migration. See "Configuring Node Manager for Whole Server Migration" for more information.
- Configured wlsifconfig.sh and tested floating IP address creation. See "Configured Whole Server Migration Floating IP Controls."

Creating the Leasing Tablespace and Active Table in the Database

To create a leasing tablespace in the Oracle database, do the following:

1. Log in to the Oracle RAC database using the SCAN address. For example:

```
sqlplus sys/password'(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=database-cluster-scan.localdomain)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=pdb1)))' as sysdba
```

2. Confirm which ASM disk group you want to use to store the leasing tablespace. For example, the following command shows that there is one ASM disk group available:

```
select group_number, name from v$asm_diskgroup;

GROUP_NUMBER NAME
------------ ------------------------------
           1 DATA
```

3. Verify the current tablespaces. For example:

```
select tablespace_name from dba_data_files;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
USERS
DEV_IAS_OPSS
DEV_IAS_IAU
DEV_STB
LARGE_DATA
LARGE_INDEX

8 rows selected.
```

4. Check the available space on the disk group. For example:

```
select name, state, total_mb, free_mb from v$asm_diskgroup;

NAME                  STATE           TOTAL_MB       FREE_MB
---------- ----------- ---------- ----------
DATA                  CONNECTED       20456          6908
```

5. Create a tablespace for the leasing table. For example:

```
create tablespace leasing logging datafile '+DATA' size 100M extent management local
uniform size 64K;
```

6. Verify the leasing table space has been created. For example:

```
select tablespace_name from dba_data_files;

TABLESPACE_NAME
------------------------------
SYSTEM
SYSAUX
USERS
DEV_IAS_OPSS
DEV_IAS_IAU
DEV_STB
LEASING
LARGE_DATA
LARGE_INDEX

9 rows selected.
```

7. Create a user account with permissions for the leasing tablespace. For example:

```
create user leasing identified by password;
grant create table to leasing;
grant create session to leasing;
alter user leasing default tablespace leasing;
alter user leasing quota unlimited on leasing;
```

8. Exit the SQL*Plus session. For example:

```
quit
```

9. Copy the **leasing.ddl** file from *WebLogic_home***/server/db/oracle/920** on one of the WebLogic Server machines. For example, from the current database directory, you can run the following command to copy over the file:

```
scp weblogic_user@ip_address:WebLogic_home/server/db/oracle/920/leasing.ddl .
weblogic_user@ip_address's password: weblogic_password
leasing.ddl
```

where

- *weblogic_user* is the WebLogic Server user account of one of the machines running a WebLogic Server.
- *ip_address* is the IP address of the machine running a WebLogic Server.
- *weblogic_password* is the password of the WebLogic Server user account.

10. From the current directory where the **leasing.ddl** file is located, log in to the newly created leasing tablespace using the SCAN address. For example:

```
sqlplus leasing/password'(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
(HOST=database-cluster-scan.localdomain)(PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=pdb1)))' as sysdba
```

11. Run the **leasing.ddl** script. You can safely ignore error messages as long as the **active** table is successfully created. For example:

```
@leasing.ddl
```

12. Verify that the **active** table has been created on the leasing tablespace. For example:

```
select table_name from user_tables;

TABLE_NAME
------------------------------------------------------------------
ACTIVE
```

Create the Leasing Multi Data Source

The leasing multi data source is the WebLogic Server connections to the Oracle RAC database instances that contain the leasing table. WebLogic Server uses this data source when migrating a managed server from one machine to another.

To create the leasing multi data source, do the following:

1. Log in to the WebLogic Remote Console.
2. From the **Edit Tree**, expand **Services**.
3. Select **Data Sources**.

   The Summary of JDBC Data Source screen is displayed.
4. Click **Create**.

   The Create a New JDBC Multi Data Source screen is displayed.
5. In the **Name** field, enter **leasing**.
6. In the **JNDI Name** field, enter **jdbc/leasing** as the JNDI name.
7. Select **Data Source Type** as **Multi Data Source**. Then from the **Algorithm Type** list, select **Failover**.
8. Click **Next**.

   The Select Targets screen is displayed.
9. In the **Clusters** table, select **All servers in the cluster**.
10. Click **Next**.

    The Select Data Source Type screen is displayed.
11. Select **Non-XA Driver**.
12. Click **Next**.

    The Add Data Sources is displayed.
13. Click **Create a New Data Source**.

    The JDBC Data Source Properties screen is displayed.
14. In the **Name** field, enter **leasing-rac1**.
15. In the **JNDI Name** field, enter **jdbc/leasing-rac1**.
16. In the **Database Type** list, select **Oracle**.
17. Click **Next**.
18. In the **Database Driver** list, select **Oracle's Driver (Thin) for RAC Service-Instance connections; Versions:Any**.

19. Click **Next**.

    The Transaction Options screen is displayed.
20. Deselect **Supports Global Transaction**. Keep One-Phase Commit selected.
21. Click **Next**.

    Connection Properties is displayed.
22. In the **Service Name** field, enter the service name of the database.
23. In the **Database Name** field, enter the database SID of the first Oracle RAC database instance. For example, rac1.
24. In the **Host Name** field, enter the database SCAN host name.
25. In the **Port** field, enter the port number of the SCAN host name.
26. In the **Database User Name**, enter leasing.
27. In the **Password** and **Confirm Password** fields, enter the leasing schema password.
28. Click **Next**.

    The Test Database Connection screen is displayed.
29. Click **Test Configuration** and do one of the following:
    - If the test failed, click **Back** to review the previous screens.
    - If the test succeeded, click **Next**.

      The Select Targets screen is displayed.
30. In the **Clusters** table, select **All servers in the cluster**.
31. Click **Finish**.

    The Add Data Sources screen is displayed.
32. Select **leasing-rac1** from the Available table and move it to the Chosen table.
33. Repeat steps 13 to 32 with the following changes:
    - In the **Name** field, enter **leasing-rac2**.
    - In the **JNDI Name** field, enter **jdbc/leasing-rac2**.
    - In the **Database Name** field, enter the database SID of the second Oracle RAC database instance. For example, rac2.
    - Select **leasing-rac2** from the Available table and move it to the Chosen table.
34. Click **Finish**.
35. Click the shopping cart. From the shopping cart, click **Commit Changes**.

Configure the Cluster for Whole Server Migration

To configure the cluster for whole server migration, do the following:

1. Log in to the WebLogic Remote Console.
2. From the **Edit Tree**, expand **Environment**. From **Environment**, select **Clusters**.

   The **Summary of Clusters** screen is displayed.
3. Select the OSM cluster.

   The b tab is displayed.

4. Select the **Migration** sub tab. Here, select the **General** sub tab.
5. In the Candidate Machines For Migration Servers tables, select the machines from the Available table that you want to use in whole server migration scenarios. Move the selected machines to the Chosen table.
6. From the **Migration Basis** list, select **Database**.
7. From the **Data Source For Automatic Migration** list, select **leasing**.
8. From the **Auto Migration Table Name**, enter **ACTIVE**.
9. Click **Save**.
10. Click the shopping cart. From the shopping cart, click **Commit Changes**.

Configure Managed Servers for Whole Server Migration

To configure managed servers for whole server migration, do the following:

1. Log in to the WebLogic Remote Console.
2. From the **Edit Tree**, expand **Environment**.
3. Select **Servers**.

   The Summary of Servers screen is displayed.
4. Select a managed server from the list.

   The Configuration tab is displayed.
5. Select the **Migration** sub tab.
6. In the Migration Configuration area, select **Automatic Server Migration Enabled**.

> **Note**
>
> This option enables node manager to automatically create and remove the floating IP addresses associated to the managed servers that it runs. It also enables node manager to migrate a managed server to another machine.

7. From the Candidate Machines area in the Available table, select the machine name that you want the managed server to primarily run on and move the machine to the Chosen table.
8. Select and move the second machine to the chosen table.

> **Note**
>
> If node manager cannot start the machine on the first machine in the chosen list, it will migrate the managed server to the second machine.

9. Select the **Services** sub tab.
10. In the Default Store area, in the **Directory** field, enter a path to a shared directory that is also accessible on the machine where the managed server can migrate to.

    For example:

    ```
    /mnt/shares/oracle/cluster/defaultdatastore/
    ```

11. Click **Save.**
12. Click the **Advanced** link below the Default Store area.
13. Deselect **Enable File Locking**.
14. Click **Save**.
15. Click the shopping cart. From the shopping cart, click **Commit Changes**.
16. Repeat this procedure for all other managed servers that can be migrated.

Testing Whole Server Migration

To test whole server migration, do the following:

1. Open a terminal on each machine running node manager.
2. Go to *domain_home***/bin** for your base domain.
3. Run the following command which starts nodemanager:

   ```
   startNodeManager.sh
   ```
4. Open a second terminal on the machine running the administration server.
5. Go to *domain_home***/bin** for your base domain.
6. Run the following command which starts the administration server:

   ```
   startWebLogic.sh
   ```
7. Log in to the WebLogic Remote Console'.
8. From the **Monitoring Tree**, select **Environment**.
9. Navigate to **Node Manager Logs**.
10. In the Node Manager Status tab, verify that the **Status** field shows the **Reachable** state.
11. Repeat steps 8 to 10 for all other machines.
12. From the Domain Structure tree, expand **Environment** and select **Servers**.

    The Summary of Servers screen is displayed.
13. Click the **Control** tab.
14. Select all managed servers that you want to start.
15. Click **Start.**
16. Click **Yes**.
17. Open a terminal on one of the machines.
18. Run the following command:

    ps -ef | grep *managed_server*

    where *managed_server* is the name of the managed server you want to test for whole server migration. For example:

    ```
    ps -ef | grep osmms1
    oracle 12785 12721 9 21:13 ? 00:01:36 /usr/java/jdk-21.0.7/bin/java ...
    ```
19. The first set of numbers in the response is the process ID (PID) of the managed server. Run the following command to terminate the managed server process:

    kill -9 *pid*

where pid is the process ID of the managed sever process you want to terminate.

20. Node manager will attempt to restart the managed server on the current machine. Wait for a few minutes, then repeat steps 18 to 19.
21. After you terminate the managed server PID a second time, go to the second machine and view the output from the terminal that you used to start node manager. Node manager should begin creating new directories for the migrating managed server, then starting up the managed server and allocating the floating IP address to the second machine. For example:

```
<INFO> <wls1412_domain> <osmms1> <Creating directory "/u01/app/oracle/
config/domains/wls1412_domain/servers/osmms1/logs">
<INFO> <wls1412_domain> <osmms1> <Creating directory "/u01/app/oracle/
config/domains/wls1412_domain/servers/osmms1/security">
<INFO> <wls1412_domain> <osmms1> <Creating directory "/u01/app/oracle/
config/domains/wls1412_domain/servers/osmms1/data/nodemanager">
<INFO> <wls1412_domain> <osmms1> <Creating directory "/u01/app/oracle/
config/domains/wls1412_domain/servers/osmms1/tmp">
...
<INFO> <Generated command - sudo /sbin/ifconfig eth0:2 192.168.56.121
netmask 255.255.255.0>
<INFO> <Successfully brought 192.168.56.121 netmask 255.255.255.0 online
on eth0:2>
<INFO> <wls1412_domain> <osmms1> <The server 'osmms1' is running now.>
```

22. Return to the WebLogic Remote Console.
23. From the **Monitoring Tree**, select **Enviroment**. From **Environment**, select **Migration**
24. Verify the status of your migrated server. For example, in the **Start Time** column, you can verify when the migration occurred. From the **Machines Migrated From** and the **Machines Migrated To** column, you can verify which machines node manager migrated the machine from and to. For more information about these columns, click the Help button in the WebLogic Server Remote console.

Migrating a Managed Server Back

To migrate a managed server back to its original location, do the following:

1. Log in to the WebLogic Remote Console.
2. From the **Monitoring tree**, expand **Environment**.
3. Select **Servers**.

   The **Summary of Servers** screen is displayed.
4. Select the migrated managed server.
5. Click **Shutdown** the **Force Shutdown Now**.
6. Click **Yes**.
7. When the managed server has shutdown, select the managed server again.
8. Click **Start**.
9. Click **Yes**.
10. Open a terminal on the first machine where the managed server had originally been located.

11. Run the following command and confirm that the managed server has begun running on the original machine:

    ```
    ps -ef | grep managed_server
    ```

    where *managed_server* is the name of the managed server you want to test for whole server migration. For example:

    ```
    ps -ef | grep osmms1
    oracle 12785 12721 9 21:13 ? 00:01:36 /usr/java/jdk-21.0.7/bin/java ...
    ```

12. From the **Monitoring tree**, select **Environment**. From **Environment**, select **Migration**.
13. Verify the status of your migrated server. For example, in the **Start Time** column, you can verify when the migration occurred. From the **Machines Migrated From** and the **Machines Migrated To** column, you can verify which machines node manager migrated the machine from and to. For more information about these columns, click the Help button in the WebLogic Server Remote console.

Installing OSM in a Clustered Environment

To install OSM in a clustered environment:

1. Start the administration server, and at least one managed server in the cluster.

   > **Note**
   >
   > The OSM installer requires that at least one managed server is running in the cluster during the installation process. The OSM Administration server configures any remaining managed servers when they are started.

2. Perform OSM installation using the procedure described in "Installing OSM" with the following modifications:
   - Run the configuration phase where you define all the required properties using the discovery script.
   - Installation phase where the installer reads the pre configured the properties and installs OSM on the Weblogic Domain.
3. After OSM is installed, restart all the servers included in the cluster.

6
Installing OSM

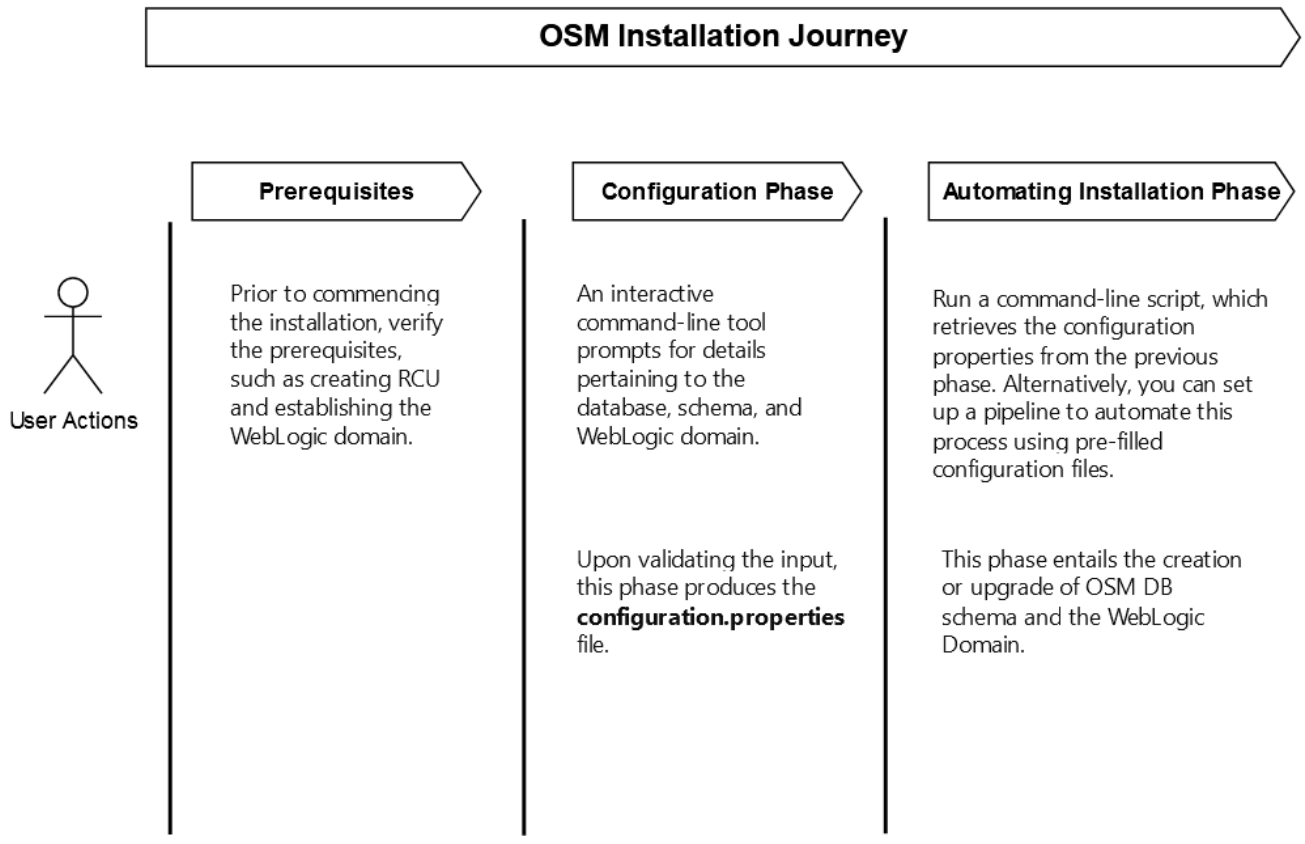
This chapter describes the OSM installation process, tailored specifically for Oracle Linux. The installation package is offered as a RedHat Package Manager (RPM) package as well as a .zip file. On Linux systems, you install the RPM package using the RPM package manager or the DNF and Microdnf tools.

About Traceability
Leveraging DNF, you can install the OSM Installer on your Linux machines. DNF installation offers comprehensive traceability, enabling you to track the installation of different OSM versions and their respective installation locations.

About the Installation Journey

The following image illustrates the installation journey.

Figure 6-1 OSM Installation Journey



Before installing OSM, read the following chapters:

- OSM Production Installation Overview

- OSM System Requirements
- Installing and Configuring the Oracle RAC Database
- Installing and Configuring the WebLogic Server Cluster

Downloading the OSM Package Installer

To download the OSM package installer:

> **Tip**
>
> If you intend to install more than one OSM instance on the same machine, Oracle recommends that you copy the libraries into a location outside *OSM_Home*.

1. Download JDK 21 from the respective website and install it, preferably in the **/usr** or the **/opt** directory.

   > **Note**
   >
   > JRE 21 is not bundled with the OSM software media pack for any operating system.

2. Download the OSM software media pack for your operating system from the Oracle software delivery website, located at:

   `https://edelivery.oracle.com/`
   and save it to a temporary directory:
3. Unzip the OSM software media pack.
4. Ensure that the Oracle Database and Oracle WebLogic Server instances that you intend to use for OSM are running.
5. For a Linux system, ensure that you have installed all required software packages, including Software Development Workstation. For information about the available software packages, see the latest *Oracle Linux Installation Guide*.

About Supported Platforms

OSM is supported on the following operating system:

- **Oracle Linux**: The OSM installer is delivered as an RPM file for Linux. Oracle products certified on Linux are also supported on RedHat Enterprise Linux due to implicit compatibility between both distributions. Oracle does not run any additional testing on RedHat Enterprise Linux products. For more information about which versions of Linux are compatible with the OSM installer, refer to the *OSM Compatibility Matrix*.

Prerequisites for Installation

Before starting the installation procedure, ensure that the following prerequisites have been satisfied:

- **Weblogic Server Clustering**: Ensure the WebLogic Server domain has been created and the required server instances and clusters have been configured.
- **Database Planning**: Ensure the required Fusion MiddleWare database schemas have been created using the Repository Creation Utility (RCU).
- **Java**: Ensure that you are using Java version 21. Refer to the *OSM Compatibility Matrix* for specific minimum update requirements. Also, ensure that you have set the **JAVA_HOME** environment variable.

Installing the OSM Installer Package

You can install the OSM Installer RPM package on Linux as an RPM or DNF file.

Regardless of the chosen tool, it is recommended that you override the default installation location. This ensures that you can easily access patches and version upgrades in the future. Additionally, overriding the installation location facilitates upgrade preparation and rollback scenarios.

You can use the OSM Installer in either a Centralized manner or a Distributed manner as described below:

- Centralized approach:
  - In the centralized approach, you need to configure a single Linux machine as the installer platform. The installer is deployed on this machine and used to target multiple environments remotely.
  - In this approach, you can streamline versioning, permissions, and related configurations for pipeline automation.
  - You can also utilize the installer state directory `-c` as a singleton, enhancing the ease of source control.
  - In this approach, Fusion Middleware installation is required to run the OSM installer. Setting up a domain, cluster, or OSM is not necessary as only Fusion Middleware is needed.

> **Note**
>
> See *OSM Compatibility Matrix* for the recommended version of Fusion Middleware installation and patch level.

- Distributed approach:
  - In the distributed approach, you can host the installer on each standalone WebLogic Server machine or admin server machine, for cluster, and target that specific environment.
  - Fusion Middleware is present due to the existing operational Weblogic server on that machine.
  - However, in this approach, version control, permissions and other configurations are not suitable for pipeline automation.

Install Using DNF
When attempting to install a new version of the OSM installer RPM package, DNF automatically handles the process of either downgrading or upgrading to the specific version.

For example:

- If OSM installer build version B2566 is already installed and an older version, say version B2559 is being installed, DNF will replace or downgrade the existing version B2566 to version B2559.
- Conversely, if build version B2559 is already installed and a newer version, say B2577 is being installed, DNF will replace or upgrade the existing version B2559 to version B2577.

To accommodate multiple versions of the OSM installer, `installroot` option can be utilized with the `dnf install` command. This allows for specifying a custom directory location. For example:

```
#install the RPM package using the dnf command
$ sudo dnf install --installroot=/path/to/desired/installationdir/ /path/to/rpm/osm-installer-7.5.0-1.el8.x86_64.rpm
```

> **Note**
>
> If the "dnf install" command shows the warning `Unable to detect release version (use '--releasever' to specify release version)`, specify the release version of the operating system you are using `--releasever` option. For Example: `dnf install --releasever=8 for Linux 8`

> **Note**
>
> Using DNF incurs a disk cost, as a new root file system is created for each custom location. Approximately 2.5 GB of disk space is required per location when using DNF.

Install Using RPM

RPM is a widely used utility for software installation on Linux systems, especially Red Hat Linux. The following is an example that illustrates how to utilize RPM to install the OSM installer:

- Access the root account, or employ the `su` command to switch to the root user on the workstation where you intend to install the software.
- Download the desired OSM package.
- Run the following command at the prompt to install the package:

```
#install with the rpm command directly
sudo rpm -ivh  --prefix=/path/to/desired/installationdir/  /path/to/rpm/osm-installer-7.5.0-1.el8.x86_64.rpm
```

> **Note**
>
> If you do not specify the prefix, then RPM will be installed in **/usr/bin** by default.

Unzipping the OSM Package

To unzip the OSM package:

1. Navigate to the directory where the OSM installation package is saved. You may have downloaded it to your user's home directory or a specific folder designated for software installations.

2. Use the `unzip` command to extract the contents of the installation package. If the package is in ZIP format, the command would look like this:

   ```
   unzip osm_installation_package.zip
   ```

Upon successful installation of the package, you will have access to the **osm-installer-*version*** folder, containing components such as:

- db-model
- libs
- scripts
- wdt
- wlsdeploy

> **Note**
>
> When you extract the OSM 8.0 installer using `sudo` (or as the `root` user), the extracted files and directories are owned by `root`. Before running any installer scripts as a non-root user, you must update permissions to ensure that the intended user can write to the installation and log directories.

Specifying Configuration Properties in the Configuration Phase

The OSM installer runs Discovery Process, which offers an interactive stage to capture the parameter settings necessary for installation activities. This process primarily centers around WLS (WebLogic Server) Domain discovery along with querying the DB to find out if it is RAC. This phase captures all the necessary data for the OSM installation.

To begin this process, you need to run the shell command **discover.sh** and provide a name for the discovery (called environment) along with a passphrase environment variable that consists of a passphrase. Throughout this process, the installer runs validations to ensure connection to both the admin server and the database server.

> **Note**
>
> The OSM installer relies on a user-provided passphrase to encrypt sensitive information such as passwords in the **configuration.properties** file. WDT is used to encrypt and decrypt these fields. The passphrase is not stored by the installer and must be provided at each installation step.

Upon successful validation, you specify the remaining OSM settings. Upon completion, a property file named **`configuration.properties`** is generated at the location you have specified in the -c option while running the discovery module. If you have not provided the location, then the property file can be found in the default location, which is **~/.osm/configuration**. You can modify the **`configuration.properties`** file. Ensure that you proceed with caution while making any changes. If you need to update the encrypted properties, then you must rerun the discovery script. You also have the option to make direct changes if you prefer that.

When handling password fields, it is crucial to not input plain text values. The discovery script prompts you for various pieces of information, including details about the database connection, WebLogic Server (WLS) admin credentials, Coherence Cluster settings, and other relevant

parameters. Run the **$OSM_INSTALLER_HOME/scripts/encrypt.sh** encryption script to convert plain text into an encrypted string. Subsequently, you should paste this encrypted string output into the **configuration.properties** file.

The OSM installer generates the following files and artifacts, which can be found in the **$OSM_CFG_HOME/configuration/environment-name** directory. If **$OSM_CFG_HOME** is not defined, then the directory is **$HOME/.osm/configuration/environment-name**:

- **osm_schema_installs** or **installer_schema_upgrades**
- **osm-wdt-app-archive.zip**
- **update_domain_output**
- **wdt_logs**
- **configuration.properties**

To begin the discovery process:

1. Navigate to the **osm-installer** folder, which becomes available after RPM, DNF or .Zip installation.
2. Locate the **scripts** folder and run either of the following commands:

> **Note**
>
> It is strongly recommended to use separate directory hierarchies for the OSM installer location and the OSM configuration root directory.

Use the following command if you are choosing to not use the `-p` parameter.

```
$ export OSM_INSTALLER_HOME=path/to/osm/installation #This is available after RPM or
DNF installation.
$ export PASSPHRASE=passphrase
$ $OSM_INSTALLER_HOME/scripts/discover.sh -n osm_env_name -c /path/to/rootdir/for/
configurations -l $OSM_INSTALLER_HOME
```

Use the following command if you are choosing to use the `-p` parameter.

```
$ export OSM_INSTALLER_HOME=path/to/osm/installation #This is available after RPM or
DNF installation.
$ export PASSPHRASE_ENV_NAME=passphrase
$ $OSM_INSTALLER_HOME/scripts/discover.sh -n osm_env_name -c /path/to/rootdir/for/
configurations -l $OSM_INSTALLER_HOME -p PASSPHRASE_ENV_NAME
```

> **Note**
>
> Here, `OSM_INSTALLER_HOME` refers to the location where the OSM installer was deployed (using RPM or DNF) or unzipped.

Use the following options with the script:

- **-n**: This is a mandatory parameter. This is the OSM environment name.
- **-c**: This is an optional parameter. This is the OSM configuration root directory. If you opt to omit this parameter, then the default configuration directory at **~/.osm/configuration** will be used.

- **-l**: This is an optional parameter. This is the OSM installer home directory. If you opt to omit this parameter, then the environment variable must be set to contain the full path of the OSM installer root directory.
- **-p**: This is an optional parameter. This is a passphrase environment variable that consists of passphrase used to encrypt and decrypt. If you omit this parameter, then the environment variable **PASSPHRASE** must be set to contain the passphrase string.
- **-h**: This parameter is for help.

Furthermore, discovery can be initiated using an existing configuration file, from which the configuration scripts can read the properties to obtain the current contents as default values. This makes it more convenient to modify configuration as the existing values can be accepted as default values until you get to the place where the values need to be changed.

The `-c` option specified during the discovery script run gives the root location for configurations. The specific configuration is stored in a subdirectory named after the environment. This setup enables a single root location to be linked to source control, supporting the discovery and configuration of multiple target environments.

> **Note**
>
> You can run the script with `-h` for more details

System Reserved Special Words

During the discovery process, certain words are reserved for screen flow control and to gracefully exit the process.

Table 6-1 System Reserved Special Words

Reserved Word	Description
Next	Skips the current section and displays the next section. Using the **Next** keyword on a section results in all configuration data from that page taking the current value from **configuration.properties** and that can be blank if the data does not already exist. For some configuration data, this will be invalid information such as the location of the DB Server during a fresh install. Improper use of **Next** can therefore result in downstream sections experiencing issues as well as the possibility of having unusable **configuration.properties**.
Back	Takes you back to the previous section.
Exit	Upon typing exit, the system prompts you to decide whether the currently captured properties should be saved or discarded.

> **Note**
>
> In the event of a validation failure, you will be prompted to enter valid input. This action triggers the display of error details, providing you with specific information to correct your input.

Specifying Configuration Properties

To specify configuration properties:

1. Initiate the discovery process as described above.
   This displays the **Welcome** section.

2. Read through the text in the Welcome section and press Enter.

   The header for the **Third Party Readme** section is displayed.

3. For the `View third party readme` property, if you enter `false`, it skips displaying the third-party related contents and the next section appears. If you enter `true`, the Readme content appears.

   ```
   Enter value for--third-party-readme (View third party readme. (default:
   true)):
   ```

4. Review the readme contents for the third-party components and then press Enter.

   The **Database Connection Information** section appears.

5. Specify the Oracle Database instance where the OSM database schema will be installed or the existing OSM schemas will be upgraded. This can be a single-instance database or an Oracle Real Application Clusters (RAC) database. If you are installing an Oracle RAC database, specify connection information for the primary database instance. Oracle RAC configuration occurs later in the installation process.

   > **Note**
   >
   > For Oracle RAC One Node, specify only the **Service Name**.

   - For the database host property, enter the IP address or DNS name of the host where the database listens for requests. For an Oracle RAC database, enter the Single Client Access Name (SCAN).

     ```
     --db-host (The Database Host. (default: null)): dbhost
     ```

   - For the database port property, enter the port where the database listens for requests.

     ```
     --db-port (The Database Port. (default: 1521)): dbport
     ```

   - For the database service property, enter the service name of the database. If the database is a pluggable database (PDB) within a container database (CDB), use the service name of the PDB. For example, the name of either the default database service or a service created specifically for OSM.

     ```
     -db-service (The Database Service Name. (default: null)): service_name
     ```

     For an Oracle RAC database instance with a remote listener (SCAN listener), you must enter both service name and SID. For other types of database, or an Oracle RAC database instance with only local listeners, you can enter either a database service name or SID, but both fields cannot be empty.

     > **Note**
     >
     > If a service is configured for OSM, all WebLogic database transactions are run against that service as expected. However, OSM jobs run by the Notification Engine are submitted to the database through the DBMS_JOB package and are not subject to any restrictions that may have been placed on the service.

- For the database system identifier property, enter the name (system identifier) of the database instance. If the database is a PDB within a CDB, use the system identifier of the CDB.

  For an Oracle RAC database instance with a remote listener, you must specify the SID and the service name. Otherwise, the WebLogic data source will not be able to override server-side load balancing. See "Listener Considerations for Oracle RAC" for a discussion of listener functionality.

  ```
  --db-sid (The Database System Identifier SID. (default:null)): sid
  ```

The **Database Administrator Credential Information** section appears.

6. Enter the credentials supplied by your Database Administrator for the following parameters:
   - For `--db-admin-username`, enter the user name for the database administrator user.
   - For `--db-admin-password`, enter the password for the database administrator user.

   > **Note**
   >
   > If you choose to connect as the **sys** user and you have not set the O7_DICTIONARY_ACCESSIBILITY database parameter to **TRUE**, append **as sysdba** to the value in the **User Name** field.

   The **Database Credential Information** section appears.

7. Enter values for the following parameters:
   - Enter the value for the OSM core schema name, in the field.

     ```
     --db-osm (The OSM core schema name (default: OSM)): osmschema
     ```

   - Enter the value for the OSM core schema password, in the field.

     ```
     --db-osm-password (The OSM core schema password. (default: null)):
     password
     ```

   - Enter the value for the OSM rule engine schema name, in the field.

     ```
     --db-rule-engine (The OSM rule engine schema name. (default: OSMRule)):
     osmschema_rule
     ```

   - Enter the value for the OSM rule engine schema password, in the field.

     ```
     --db-rule-engine-password (The OSM rule engine schema password
     (default: null)): password
     ```

   - Enter the value for the OSM report schema name, in the field.

     ```
     --db-report (The OSM report schema name. (default: OSMReport)):
     osmschema_report
     ```

- Enter the value for the OSM report schema password, in the field.

  ```
  --db-report-password (The OSM report schema password. (default: null)):
  password
  ```

The **Database Schema Tablespaces** section appears

8. Enter values for the following parameters to allocate space for the OSM database schemas.
   - Enter the value for the name of the default tablespace, in the field.

     ```
     --db-default-tablespace(The name of the default tablespace
     [{"name":"OSM","availableSpaceMB":12677},
     {"name":"SYSAUX","availableSpaceMB":137}] (Required TableSpace : 35MB)
     (default: OSM)):
     ```

   - Enter the value for the name of the temporary tablespace, in the field. This specifies the temporary tablespace for the database schema.

     ```
     --db-temp-tablespace(The name of the temp tablespace
     [{"name":"TEMP","availableSpaceMB":99}] (default: TEMP)):
     ```

   - Enter the value for the name of the model data tablespace, in the field. This specifies the model data tablespace for the database schema.

     ```
     --db-model-data-tablespace(The name of the model data tablespace
     [{"name":"OSM","availableSpaceMB":12677},
     {"name":"SYSAUX","availableSpaceMB":137}] (Required TableSpace : 35MB)
     (default: OSM)):
     ```

   - Enter the value for the name of the model index tablespace, in the field. This specifies the model index tablespace for the database schema.

     ```
     --db-model-index-tablespace (The name of the model index tablespace
     [{"name":"OSM","availableSpaceMB":12677},
     {"name":"SYSAUX","availableSpaceMB":137}] (Required TableSpace : 35MB)
     (default: OSM)):
     ```

   - Enter the value for the name of the order data tablespace, in the field. This specifies the order data tablespace for the database schema.

     ```
     --db-order-data-tablespace (The name of the order data tablespace
     [{"name":"OSM","availableSpaceMB":12677},
     {"name":"SYSAUX","availableSpaceMB":137}] (Required TableSpace : 35MB)
     (default: OSM)):
     ```

   - Enter the value for the name of the order index tablespace, in the field. This specifies the order index tablespace for the database schema.

     ```
     --db-order-index-tablespace (The name of the order index tablespace
     [{"name":"OSM","availableSpaceMB":12677},
     {"name":"SYSAUX","availableSpaceMB":137}] (Required TableSpace : 35MB)
     (default: OSM)):
     ```

For each tablespace, you are shown how much space you have available and how much space is required. For more information, see "Tablespaces."

The **Database Schema Partition Information** section appears.

9. The information gathered here is used to determine whether the database schema is created using partitions.

   - Enter the value for the database partition property. The default value for this is `true`. If you choose `false`, this section will be skipped.

     ```
     --db-partition (Use Oracle Partitioning features for optimal
     performance in high volume production environment. (default: true)):
     Usage: partitionProperties
     ```

   - Enter the value for database partition size property. This is the number of orders that will be allowed in a partition. The default value for this is 20000000.

     ```
     --db-partition-size (The size of each partition. (default: 20000000)):
     ```

   - Enter the value for the database subpartition count property. This is the number of sub-partitions allowed in a partition. The default value for this is 32.

     ```
     --db-subpartition-count (The total number of subpartition. Default is
     32 ):
     ```

     After installation, you can change the values that you selected during the installation process by updating the `--db-partition-size` and `--db-subpartition-count` OSM database parameters. However, updates to these parameters do not affect existing partitions.

     > **Note**
     >
     > For more information about partitioning, see *OSM System Administrator's Guide*.

   > **Important**
   >
   > Oracle strongly recommends partitioning in all production deployments or production test environments, particularly those with high order volumes or any volume of large or complex orders. If you choose not to partition your OSM schema, it could be expensive to later reverse your decision. Changing a non-partitioned schema that has accumulated a large volume of data to a partitioned schema involves time-consuming and resource intensive export/import.

The **Database Timezone Information** section appears.

10. The OSM Database Timezone must be set to Non-Daylight Saving time. You can inform OSM of this timezone as an offset in the value of seconds. Enter the value for the database timezone offset in seconds.. The default value is set to -28800.

    ```
    --db-timezone-offset-seconds (The database time zone offset in second
    (default: -28800)):
    ```

    > **Note**
    >
    > For more information on timezone settings, see "Configuring Time Zone Settings in the Database" or *OSM System Administrator's Guide*.

    The **Weblogic Server Connection Information** section appears.

11. Do the following:

    a. For the weblogic administrator server host property, enter the name or IP address of the machine where WebLogic is installed.

       ```
       --weblogic-admin-server-host (The Weblogic Admin Server Host. (default:
       null)): wlhost
       ```

    b. For the weblogic administrator server port property, enter the port where WebLogic is operating. .

       ```
       --weblogic-admin-server-port (The Weblogic Admin Server Port. (default:
       null)): wlport
       ```

    c. For the weblogic administrator user name property, enter the name of the WebLogic administrator.

       ```
       --weblogic-admin-user-name (The Weblogic Admin User name. (default:
       null)): wluser
       ```

    d. For the weblogic administrator user password property, enter the password of the WebLogic administrator.

       ```
       --weblogic-admin-user-password (The Weblogic Admin User Password.
       (default: null)): password
       ```

    e. For the connection to weblogic using SSL property, enter `true` to use an SSL connection to the WebLogic admin server. The server's SSL port must be enabled to use this feature. By default, a non-SSL connection is used.

       ```
       --weblogic-ssl-enabled (Connect to Weblogic via SSL. (default: false)):
       true
       ```

    f. If you are using SSL, for the weblogic port, enter the port number of the WebLogic admin server.

       ```
       --weblogic-admin-server-ssl-port(The Weblogic SSL Port. (default:
       null)): sslport
       ```

g. If you are using SSL, for the keystore file, specify the location of the key store file required for the SSL connection by typing the full path and directory.

```
--ssl-keystore-file(SSL Keystore File Name. (default: null)):
keystore_file_path
```

The keystore file must exist locally in the machine from where the OSM installer is running.

h. Along with the SSL keystore file location, provide the password for the SSL keystore file.

```
--ssl-keystore-file-password (SSL Keystore Password. (default: null)):
password
```

The **WebLogic Server (WLS) Deployment Target** section appears.

12. For the weblogic deployment target, you can enter the values `CL`, or `AdminServer`. Here, `CL` is the cluster name and it can be anything based on the name provided during domain creation. The screen will detect the cluster name and populate here and you have to enter the value accordingly.

- If you created a single WebLogic admin server, enter `AdminServer`.
- Select the appropriate cluster from the list detected.

```
--weblogic-deployment-target(Cluster/Stand-alone Server (detected:
CL,AdminServer, current: AdminServer)): CL
```

13. If you are performing an upgrade and there are differences in the existing JDBC configuration and the proposed JDBC configuration, then the **Database Configuration Comparison** section is displayed.

If you specified an Oracle RAC database, the **Oracle Real Application Clusters Configuration** section is displayed. Go to step 15.

If you have specified a WebLogic cluster, the **WebLogic Cluster Web Service Request Configuration** section is displayed. Go to step 22.

14. This section displays the existing JDBC configuration and the proposed new JDBC configuration. You can choose to replace the existing JDBC Data Source Configuration with new connection information. To do that, enter `true` for the existing weblogic server data source replacement property. If you do not want to, then enter `false`:

```
--existing-wls-ds-replace (Do you want to replace the existing JDBC data
source configuration with the new connection information. (default:
true)): false
```

```
 Enter true or false to continue or enter back to change the database connection
information.
```

- Below is the new configuration:

```
[{"name":"ORCHDB2","is_rac":"Yes","host":"dbhost","port":"dbport","servi
ce":"servicename","sid":null,"user":"osmuser"}]
```

> **Note**
>
> The OSM installer does not perform any data migration. Therefore, if you select this option, all configuration parameters will be reset to their defaults.

- Below is the existing configuration:

```
[{"name":"--","is_rac":"Yes","host":"dbhost","port":"dbport","service":"
servicename","sid":"sid1","user":"osmuser"},
{"name":"--","is_rac":"Yes","host":"dbhost","port":"dbport","service":"s
ervicename","sid":"sid2","user":"osmuser"}]
```

15. If you specified an Oracle Real Application Clusters (Oracle RAC) database for the primary database instance, the **Oracle Real Application Clusters Configuration** section is displayed.

    Otherwise, go to step 21.

16. Do one of the following:

    - If the installer has detected that the specified database instance is an Oracle Real Application Clusters (RAC) instance, you can automatically configure additional Oracle RAC database instances for either load balancing or failover. The value for the use Oracle RAC property, in that case, should be `true`.

      ```
      --use-oracle-rac (Use Oracle RAC: (default: true):
      ```

    - If you want to configure additional Oracle RAC database instances for load balancing or failover, enter the value `true` for the use Oracle RAC property.

    - ```
      --use-oracle-rac (Use Oracle RAC: (default: true):
      ```

17. Do one of the following:

    a. In the **Configure WebLogic JDBC Data Sources** section, do the following:

       - If you want to use an additional RAC Database instance now, enter the value `now`, for the configuring Weblogic JDBC data sources property.

         ```
         --rac-config (Configure Weblogic JDBC data Sources.
         now, later (default: now):
         ```

         The installer preconfigures the database connections in WebLogic server.

       - If you want to manually use an additional RAC database, after installation completes, enter the value `later`, for the configuring Weblogic JDBC data sources property.

         ```
         --rac-config (Configure Weblogic JDBC data Sources.
         now, later (default: now):
         ```

         You can add more Oracle RAC database instances manually after the installation. See "Manually Configuring Additional Data Sources for an Oracle RAC Instance" for configuration details.

    b. In the **RAC operation mode** section, you can choose one of the following options:
         ```

- **Load Balancing (Active Active) - Order and Service Management WebLogic Cluster Installations**

  The installer groups the WebLogic Server instances according to the number of Oracle RAC database instances. Each group is configured to a separate Oracle RAC database as the primary data source, and the remaining Oracle RAC database instances as secondary data sources.

  This option is available only if OSM is deployed to a WebLogic cluster.

  For this option, enter the value `load_balance` for the RAC operation mode property.

  ```
  --rac-operation-mode (RAC operation mode:
   load_balance, failover (default: load_balance):
  ```

- **Failover (Active Passive)**

  The installer configures multi data sources and data sources according to the number of Oracle RAC database instances. The first data source of each multi data source connects to the primary database instance specified in step 5, and the subsequent data sources connect to other database instances to be specified. This option preconfigures the database connections in WebLogic for warm standby.

  For this option, enter the value `failover` for the RAC operation mode property.

  ```
  --rac-operation-mode (RAC operation mode:
   load_balance, failover (default: load_balance):
  ```

c. In the **Listener Configuration** section, do one of the following things:

- **Remote Listener (SCAN Listener)**

  To choose this, enter the value `remote` for the listener configuration property.

  ```
  --listener-configuration (Listener Configuration:
   remote, local (default: remote): local
  ```

- **Local Listeners**

  See "Listener Considerations for Oracle RAC" for information about listener functionality.

  To choose this, enter the value `local` for the listener configuration property.

  ```
  --listener-configuration (Listener Configuration:
   remote, local (default: remote): local
  ```

18. For a standalone setup, enter the following values for the Configuring Weblogic JDBC Data Sources and Listener Configuration parameters:

```
Enter value for --rac-config (Configure Weblogic JDBC data Sources.
now, later (default: now):
Usage:

RAC operation mode: failover
Enter value for --listener-configuration (Listener Configuration:
 remote, local (default: remote): local
```

If you entered the values, `--use-oracle-rac true`, `--rac-config now`, and `--listener-configuration remote`, the **Oracle Real Application Clusters Instances - Remote Listener (SCAN Listener)** section appears.

19. In the **Oracle Real Application Clusters Instances - Remote Listener (SCAN Listener)** section, do one of the following:

    - If you selected the remote listener option:

      a. For **SCAN Address**, you can modify the SCAN address if required. For example, you may need to do this if you entered an incorrect address for the *db_host* parameter in step 18 C.

      b. For the *db_host* parameter, you can modify the SCAN port if required. For example, you may need to do this if you entered an incorrect port in the **Port** field in step 18 C.

      c. Enter the value of the Database Service Name as you did for the primary instance and a unique SID for each additional Oracle RAC instance. For a container database, use the PDB service name and the CDB SID.

      ```
      --rac-remote-db-service (The Database Service Name. (default:
      remote-service)):
      ```

      d. If you did not specify the SID of the primary Oracle RAC instance, do so now by entering the value for the Database System Identifier property. For a container database, use the CDB SID.

      ```
      --rac-remote-db-sid (The Database System Identifier (SID).
      (default: null)): sid1
      ```

20. If you entered the values, `--use-oracle-rac true` and `--rac-config now` and `--listener-configuration local`, the following sections appear:

    - **Additional RAC Count**
      Specify the number of additional Oracle RAC instances:

      > **Note**
      >
      > At least two RAC instances should be specified to continue.

      ```
      --no-of-additional-rac (Enter the number of additional RAC Instances
             to be added (default: 2)):
      ```

    - **Oracle RAC Instances**
      Specify additional Oracle RAC instances. The instances must be in the same Oracle RAC database as the primary instance.

      a. The host and port: Each row must use a different combination of host and port. For example, if you use the same host, you must use a different port.

      For the RAC Local Database Host property, enter the value.

      ```
      --rac-local-db-host(The Database Host. (default: dbhost)):
      ```

For the RAC Local Database Port property, enter the value.

```
--rac-local-db-port(The Database Port. (default: 1521)):
```

b. The service name or SID: Each row must specify either the same service name as the primary instance or a unique SID. Rows can also specify both service name and SID. Specify the same fields as you did for the primary instance. For example, if you only specified the SID for the primary instance, specify unique SIDs for the additional instances. For a container database, use the PDB service name and the CDB SID.

   For the RAC Local Database Service property, enter the value.

```
--rac-local-db-service(The Database Service Name. (default:
servicename)):
```

   For the RAC Local Database System Identifier property, enter the value.

```
--rac-local-db-sid(The Database System Identifier (SID). (default:
null)): sid1
```

All instances must be in the same database.

21. If you selected a WebLogic cluster, the **WebLogic Cluster Web Service Request Configuration** section is displayed. Update the following parameters:

    - For the Front End Host property, enter the value.

```
--front-end-host(Frontend Host. (default: host)):
```

    - For the Front End http Port property, enter the value.

```
--front-end-http-port(Frontend HTTP Port. (default: port)):
```

    - For the Front End https Port property, enter the value.

```
--front-end-https-port(Frontend HTTPS Port. (default: 0)):
```

    These are for Web Service HTTP requests. The installer automatically populates these fields with the values configured for your WebLogic domain, proxy server, or load balancer.

    If the WebLogic default values of **<blank>**and **0** appear, update the fields with the correct values. You must update these values to proceed with the installation.

    For information about updating these fields again after installation, see the discussion about how to configure HTTP settings for a cluster in the WebLogic Remote Console Help.

22. If you specified a single-instance database in the Database Server Connection Information section, the Database Connection Pool Information section is displayed. Go to step 23.

    If you specified an Oracle RAC database with its SID, the Oracle Real Application Clusters Configuration section is displayed. Go to step 15.

    If you have specified a WebLogic cluster, go to step 25.

    The **Database Connection Pool Information** section is displayed.

23. If you are not sure how to size the pool at this time, use the default settings. These settings can be tuned later from the WebLogic Remote console. Do the following:

a. For the Initial Capacity property, enter the number of database connections initially reserved in the WebLogic connection pool.

```
--init-capacity(Initial Capacity - The number of connections created
when the server starts (0 or more). (default: 15)):
```

> **Note**
>
> In an Oracle RAC configuration, the initial capacity is set to 0 (read-only).

b. For the Maximum Capacity property, enter the maximum number of database connections reserved in the WebLogic connection pool.

```
--max-capacity(Maximum Capacity - The upper limit for the number of
database connections (14 or more). (default: 54)):
```

c. For the Capacity Increment property, enter the number of connections added when the connection pool maintained by the WebLogic Server is exhausted.

```
--capacity-increase(Capacity Increment - The number of connections that
will be created when all existing connections are in use (1 or more).
(default: 1)):
```

> **Note**
>
> In an Oracle RAC configuration, this connection pool information will be shared.

The **JMS Store Information** section is displayed.

24. For the JMS Store Information property, accept the default setting of `file` or enter the alternate `jdbc`.

```
--jms-store (Candidates: jdbc, file(default: file):
```

If you choose JMS File Store, OSM will use the default WebLogic file-based persistent store as the JMS store. After the installation is complete, Oracle recommends that you configure one custom file store for each managed server and JMS server. For more information, see "Creating and Configuring Persistent File Stores."

While filestores provide better performance than JDBC stores, the benefit of JDBC stores is that online database backups can obtain consistent snapshots of both OSM data and JMS messages. However, there is currently no mechanism for consistent backup of JDBC stores and transaction logs. For more information about backup strategies, see OSM System Administrator's Guide.

25. If OSM is being deployed to a WebLogic cluster, the **WebLogic Coherence Cluster Configuration** section is displayed.

If OSM is being deployed to a single WebLogic server, the **OSM Administrator and Deployment Credentials** section is displayed. Go to step 27.

26. Accept the settings for the WebLogic coherence cluster that the installer has detected, or create a new coherence cluster by modifying the default values.

    Enter a value for the Unicast Listen Port property.

    ```
    --unicast-port(Unicast Listen Port: (default: 17001):
    ```

    Enter a value for the Well Known Address property.

    ```
    --well-known-address(IP address for the server MS2 where OSM is targeted.
    (default: host):
    ```

    Enter a value for the Well Known Address property.

    ```
    --well-known-address(IP address for the server MS1 where OSM is targeted.
    (default: host):
    ```

    The coherence cluster name is generated using the pattern: **osmCoherenceCluster***N*, where *N* is a number generated by the OSM installer. Ensure that the **IP Address** value corresponds to the IP address or machine name of the WebLogic Server where the coherence cluster is running.

    > **Note**
    >
    > Oracle recommends using unicast cluster messaging mode. The installer uses unicast mode and does not allow you to use the installer to change the mode to multicast. Even if the target WebLogic cluster is a member of a coherence cluster that uses multicast mode, the installer modifies the cluster messaging mode to unicast. For information about using WebLogic to change the cluster messaging mode to multicast, see "Configuring a Multicast IP Address for the Cluster Messaging Mode."

    The **OSM Administrator and Deployment Credentials** section is displayed.

    > **Note**
    >
    > If you get the "Well known address with listen address xx.xx.xx.xx is not valid or not reachable" error, launch the installer with a user who has root privileges.

27. The user names and passwords you provide will be used to create initial user accounts with administrator privileges and access to Oracle Communications Design Studio (the deployment tool). Do the following:

    a. For the OSM Administrator Username parameter, enter a username, or use the default `admin`.

    ```
    --osm-admin-username (Admin User Name - User account with Administrator
    privileges. (default: admin)):
    ```

b. For the OSM Administrator Account Password parameter, enter a password for the OSM Administrator user. The default is `null`.

```
--osm-admin-password (Password - The password for the Administrator
user account. (default: null)):
```

c. For the OSM Deploy Administrator User parameter, enter a user name or use the default `deployAdmin`. Design Studio uses this user to deploy cartridges to OSM.

```
--osm-deploy-admin-username (Deploy Admin User Name - User account with
privileges to deploy. (default: deployAdmin)):
```

d. For the OSM Deploy Administrator User Password parameter, enter a password for the Deploy Admin user.

```
--osm-deploy-admin-password (Password - The password for the Deploy
Admin user account. (default: null)):
```

e. Press Enter.

The **OSM WebLogic User Account Passwords** section is displayed.

28. You use this section to create passwords for the standard users that are created for the application. The **Automation User Name** and **OSM Core User Name** are provided in this section for reference only and are not editable. The passwords you enter must meet the password requirements for your WebLogic domain. Do the following:

a. For the OSM Automation User Password parameter, enter the password for the oms-automation user. This is the internal automation user, used for processing OSM automation and email notifications.

```
--osm-automation-user-password (Automation User Password - The password
for the automation user account. (default: null)):
```

b. For the OSM Core User Password parameter, enter the password for the oms-internal user. It is used for internal processing when an operation must be performed on behalf of the application rather than on behalf of the user.

```
--osm-core-user-password (OSM Core User Password - The password for the
OSM core user account. (default: null)):
```

c. For the OSM Metrics User Password parameter, enter the password for the oms-metrics user.

```
--osm-metrics-user-password (OSM Metrics User Password - The password
for the OSM metrics user account. (default: null)):
```

Press Enter.

The **WebLogic Coherence Cache Configuration** section is displayed.

29. (Optional) To customize the default Coherence Cache Configuration, do the following:

a. If you want to customize the Coherence Cache Configuration, enter the value `true`. OSM will use the default Coherence Cache Configuration for the value `false`.

```
--use-custom-coherence-cache-config (Use Custom Coherence Cache
    Configuration. (default: false)):
```

b. Provide the custom Coherence Cache Configuration file path if `--use-custom-coherence-cache-config` is `true`. The sample file is present in the osm-installer-version/samples directory, **$OSM_INSTALLER_HOME/samples/osm-coherence-cache-config.xml**.

For more details, refer to: Configuring and Monitoring Coherence Threads

> **Note**
>
> Do not modify the sample configuration file directly inside **oms.ear**, this can result in inconsistency if not done properly.

```
--coherence-cache-config-file-path (The Coherence Cache Configuration
File
    Path):  /path/to/custom/config.xml
```

The file validation will happen if using custom configuration and next, the **OSM Server Session Information** section is displayed.

> **Note**
>
> You must preserve the Coherence Cache Configuration file and ensure it is provided to all subsequent installer runs for this environment (usually for OSM upgrade). The installer remembers the last provided location as default, but you can change that if needed.

30. The information in this section is used to configure your OSM user sessions. Do the following:

a. For the Session Timeout parameter, enter the time in minutes that Order Management web client and Task web client sessions remain active.

```
--osm-session-timeout (Session Timeout - The period of time a user
session can remain idle before it expires. range (1-1440 minutes).
(default: 45)):
```

b. For the Server Domain Suffix parameter, enter the domain suffix for the computers on which the OSM server will run.

```
--osm-server-domain-suffix (Server Domain Suffix - The Domain Suffix
(e.g. sample.com) for the computers the Order and Service Management
Server will run on. (default: oracle.com)):
```

c. For the Landing Page parameter, select the first page that Task Web Client users will see after login.

```
--osm-landing-page (Landing Page - Specifies the first page that web
client users see after logging in.
 Candidates: worklist, about, home, query (default: worklist):
```

d. The **Order and Service Management Remarks and Attachment Information** section is displayed.

31. The information in this section is used to configure the text remarks and file attachments that users can add to OSM orders. Do the following:

a. For the Maximum Attachment Size parameter, enter the maximum attachment size in MB that can be appended to a remark.

```
--osm-max-attachment-size (Maximum Attachment Size - The maximum size
of a single attachment, range (1-100 MB). (default: 3)):
```

b. For the Remark Change Timeout parameter, enter the length of time in hours that a remark can be edited before it becomes read-only.

```
--osm-remark-change-timeout (Remark Change Timeout - The length of time
before a remark becomes read-only (-1 or more hours). (default: 1)):
```

c. Press Enter.

The **Order and Service Management Notification Emails** section is displayed.

32. The information in this section is used to configure the email notifications for OSM. Do the following:

a. For the Notification Email Server parameter, enter the DNS name or IP address of your email server.

```
--osm-notification-email-server (Notification Email Server - The DNS
Name or IP address of your email server. (default: 127.0.0.1)):
```

b. For the Notification Email Server Port parameter, enter the port on which the email server is listening.

```
--osm-notification-email-server-port (Notification Email Server Port -
The port of your email server. (default: 993)):
```

c. For the Admin Email Address parameter, enter the OSM Administrator's email address.

```
--osm-admin-email-address (Admin Email Address - The email address of
the Order and Service Management admin. (default: null)):
```

d. Press Enter.

The **Task Processor Configuration** section is displayed.

33. The information in this section is used to control the rule and delay task evaluation. You can change these settings at any time after installing OSM. See *OSM System Administrator's Guide* for information about changing these values in **oms-config.xml**.

Do the following:

a. For the Task Processor Interval parameter, enter the number of seconds between task processor polls.

```
--osm-task-processor-interval (Task Processor Interval - The interval
between task processor polls, range (1 - 60 seconds). (default: 5)):
```

b. For the Maximum Rule Processor Count parameter, enter the maximum number of rule task processors used to evaluate rules.

```
--osm-max-rule-count (Maximum Rule Processor Count - The maximum number
of rule task processors, range (1 - 50). (default: 1)):
```

c. For the Maximum Delay Processor Count parameter, enter the maximum number of delay task processors used to evaluate delays.

```
--osm-max-delay-processor-count (Maximum Delay Processor Count - The
maximum number of delay task processors, range (0 - 50). (default: 1)):
```

> **Note**
>
> The total number of processors will be adjusted automatically at run time to exceed not more than 10% of the connection pool size. For a non-production environment, you can use the default values.

Installing DB Schema and OSM

In the installation phase, you carry out the following tasks:

1. Reading the pre-configured property values and facilitating the creation or upgradation of the OSM database schema.
2. Configuring the WebLogic domain for OSM and deploying the OSM application.

> **Note**
>
> The installation will be in the online mode, for which you should only have the AdminServer up and running.

Installing DB Schema and OSM Together
Use the following command if you are choosing to not use the `-p` parameter.

Set the environment variable:

```
$ export OSM_CFG_HOME=/path/to/rootdir/for/configurations
$ export FMW_HOME=/path/to/FMW_HOME
$ export PASSPHRASE=passphrase
```

Run the following command for automating the DB schema creation and OSM deployment in WebLogic domain:

```
$ $OSM_INSTALLER_HOME/scripts/configOSM.sh -n osm_env_name  -c $OSM_CFG_HOME -
l $OSM_INSTALLER_HOME -f $FMW_HOME
```

Use the following command if you are choosing to use the `-p` parameter.

Set the environment variable:

```
$ export OSM_CFG_HOME= /path/to/rootdir/for/configurations
$ export FMW_HOME=/path/to/FMW_HOME
$ export PASSPHRASE_ENV_NAME=passphrase
```

Run the following command for automating the DB schema creation and OSM deployment in WebLogic domain:

```
$ $OSM_INSTALLER_HOME/scripts/configOSM.sh -n osm_env_name -c $OSM_CFG_HOME -
l $OSM_INSTALLER_HOME -f $FMW_HOME -p PASSPHRASE_ENV_NAME
```

You can run the script with `-h` for more details. This script runs the OSM Domain Installer, which creates or upgrades the OSM database schema and also configures the Weblogic domain in online mode. Make sure the Admin server is up and running and all the managed servers are shutdown. Run the script with `-h` for more details.

Installing DB Schema and OSM in Separate Steps
You also have the option to install the DB schema and OSM Domain in two separate steps. If one of the components, such as the database, is installed successfully but the Weblogic server failed for some reason, then you can rerun only the Weblogic server component to install it.

Use the following commands if you are choosing to not use the `-p` parameter.

Set the following environment variables:

```
$ export OSM_CFG_HOME=/path/to/rootdir/for/configurations
$ export FMW_HOME=/path/to/FMW_HOME
$ export PASSPHRASE=passphrase
```

To install the OSM DB schema, run the following command:

```
$ $OSM_INSTALLER_HOME/scripts/configDB.sh -n osm_env_name -c $OSM_CFG_HOME -
l $OSM_INSTALLER_HOME
```

You can run the script with `-h` for more details

To install or upgrade OSM in WebLogic Domain, run the following command:

```
$ $OSM_INSTALLER_HOME/scripts/configDomain.sh -n osm_env_name -c $OSM_CFG_HOME -
l $OSM_INSTALLER_HOME -f $FMW_HOME
```

You can run the script with `-h` for more details

Use the following commands if you are choosing to use the `-p` parameter.

Set the following environment variables:

```
$ export OSM_CFG_HOME=/path/to/rootdir/for/configurations
$ export FMW_HOME=/path/to/FMW_HOME
$ export PASSPHRASE_ENV_NAME=passphrase
```

To install the OSM DB schema, run the following command:

```
$ $OSM_INSTALLER_HOME/scripts/configDB.sh -n osm_env_name -c $OSM_CFG_HOME -
l $OSM_INSTALLER_HOME -p PASSPHRASE_ENV_NAME
```

To install or upgrade OSM in WebLogic Domain, run the following command:

```
$ $OSM_INSTALLER_HOME/scripts/configDomain.sh -n osm_env_name -c $OSM_CFG_HOME -
l $OSM_INSTALLER_HOME -f $FMW_HOME -p PASSPHRASE_ENV_NAME
```

If you experience any issues with installation, refer to the "Troubleshooting OSM Installation Problems".

Configuring and Monitoring Coherence Threads

To configure threads counts to ensure performance, you can tune these threads in the osm-coherence-cache-config.xml file. Oracle recommends that you customize the osm-coherence-cache-config.xml file for tuning procedures as described in "OSM Pre-Production Testing and Tuning."

To set these values, do the following:

1. Get the sample configuration file from `$OSM_INSTALLER_HOME/wlsdeploy/applications/security.gar/META-INF/osm-coherence-cache-config.xml`.
2. Make a copy of the configuration file in another directory.
3. Open the copied osm-coherence-cache-config.xml file and set the osm-invocation thread-count to a value as follows:

```
<invocation-scheme>
     <scheme-name>osm-invocation</scheme-name>
   <service-name>osm-invocation</service-name>
     <thread-count>thread_count</thread-count>
     <autostart>false</autostart>
</invocation-scheme>
```

> **Note**
>
> Where `thread_count` is determined based on the following calculation 7+((the number of Managed Servers-1)*3). For example, clusters with 2, 4, 6, or 16 managed servers would require 10, 16, 22 and 52 threads respectively.

4. Set the osm-distributed thread-count to a value between 10 and 15 (the default is 4).

```
<distributed-scheme>
    <scheme-name>osm-distributed</scheme-name>
    <service-name>DistributedCache</service-name>
    <thread-count system-
property="tangosol.coherence.distributed.threads">thread_number</thread-
count>
    <backing-map-scheme>
       <local-scheme>
            <scheme-ref>osm-backing-map</scheme-ref>
       </local-scheme>
    </backing-map-scheme>
  <autostart>true</autostart>
</distributed-scheme>
```

> **Note**
>
> Where `thread_number` is a value between 10 and 15.

5. Save the changes and provide this file path while running the script `$OSM_INSTALLER_HOME/scripts/discovery.sh` for the screen **WebLogic Coherence Cache Configuration**.

7

Performing OSM Post-Installation Tasks

This chapter describes Oracle Communications Order and Service Management (OSM) post-installation tasks.

OSM Client Configuration Post-Installation Tasks

The following sections provide post-installation instructions relating to the OSM Order Management web client and the OSM Task web client.

> **Note**
>
> If there is an upgrade to OSM 7.5.0.0.1, it is normal to see multiple entries for the cartridge management WS application in the WebLogic Remote console. However, only one of those should be active with all others showing status as retired. You can delete the retired application versions at your convenience.

Enabling Graphical Display on UNIX or Linux Systems

To display graphical representations such as orchestration plans on UNIX or Linux systems, OSM requires an Xserver such as Xvfb (X virtual framebuffer), an X11 server that is available for most UNIX or Linux platforms.

If you are running multiple instances of OSM on the same system, set up a dedicated Xvfb server for each instance.

To set up dedicated Xvfb servers for multiple instances of OSM:

1. Start one instance of the Xvfb server on display 1:

   ```
   Xvfb :1
   ```

2. Set the value of the DISPLAY environment variable in the first WebLogic server domain instance to 1:

   ```
   DISPLAY=localhost:1
   ```

3. Start the first WebLogic server domain instance.
4. Repeat steps 1 to 3 for each additional OSM instance running on the same system.

> **Note**
>
> You must increment the Xvfb server display and the DISPLAY environment variable by one for each additional OSM instance. For example, the second instance would have a value of 2 for the Xvfb server display and the DISPLAY environment variable.

Connection, File Store, and Thread Configuration Post-Installation Tasks

The following sections include post-installation tasks for configuring connections, file stores, and threads.

Customizing OSM Run-Time Parameters

After installation, you may want to change OSM run-time parameters. For example, you can increase the number of rows in a worklist, or update the OSM administrator's e-mail address.

OSM run-time parameters are configured in the **oms-config.xml** file. You can access this file in the *domain_home* directory.

For details, see the chapter describing the **oms-config.xml** file in *OSM System Administrator's Guide*.

Preventing Connection Timeout Issues During Cartridge Deployment

When deploying large cartridges to an OSM system with many managed servers, you may receive cartridge deployment timeout errors. To resolve this error, increase the cartridge deployment timeout setting. The default cartridge deployment timeout setting is 600 seconds (10 minutes), but you can increase this value to a maximum of 3600 seconds (60 minutes).

To modify the default cartridge deployment timeout setting, change the value for the **CartridgeDeploymentTransactionTimeout** parameter in the *domain_home*/**oms-config.xml** file on each machine in the cluster. For example:

```
<oms-parameter>
<oms-parameter-
name>com.mslv.oms.cartridgemgmt.DeployCartridgeMDB.CartridgeDeploymentTransactionTimeout<
/oms-parameter-name>
<oms-parameter-value>3600</oms-parameter-value>
</oms-parameter>
```

For more information about changing this **oms-config.xml** parameter, see *OSM System Administrator's Guide*.

To override this default setting for a specific cartridge, you can specify this parameter as an OSM Cartridge Management Variables in Design Studio. You must use the full parameter name.

```
com.mslv.oms.cartridgemgmt.DeployCartridgeMDB.CartridgeDeploymentTransactionTimeout
```

For more information, see the Design Studio Help.

> **Note**
>
> If an order has stopped processing due to a timeout, the order must be terminated. It will not resume processing even after the timeout value has been increased.

Configuring OSM JDBC Connections

The OSM installer creates JDBC connections that enable WebLogic communication to the Oracle RAC database instances.

The **Statement Timeout** value specifies the amount of times before the database terminates SQL statements that are taking too long to complete. This value is the WebLogic Server domain Java transaction API (JTA) timeout value plus 2 seconds giving the database enough time to react to a JTA timeout. The **oracle.jdbc.ReadTimeout** value stops the database from reading a socket after the Statement Timeout. These values should not be modified without making corresponding changes to each other. For more information about setting the JTA timeout value, see "Preventing Connection Timeout when Using a Remote Database."

The **Statement Cache Size** value determines the number of SQL statements that can remain in the statement cache. In a production environment, you can start this value at 30 and increasing the value only if you cannot improve parse ratios by tuning the cursors. For a development environment, the default setting of 10 is enough.

To configure the OSM JDBC Connections, do the following:

1. Log in to the WebLogic Server Remote Console.

   The WebLogic Remote Console is displayed.
2. Click **Edit Tree**.
3. Select **Services**. From **Services**, select **Data Sources**.

   The Summary of JDBC Data Sources screen is displayed.
4. Select an OSM JDBC data source. The OSM JDBC data source are as follows:

   osm_pool_*sid*_**group**_*y*

   where *sid* is the Oracle RAC database instance system identifier and *y* is the group letter.

   The Configuration tab is displayed
5. Click the **Connection Pools** tab.
6. Click **Advanced**.
7. In the **Properties** field, add the following:

   oracle.jdbc.ReadTimeout=*value*

   where *value* is the **Statement Timeout** value converted to milliseconds plus 2000 milliseconds.

   For example, if the Statement Timeout value is 602 seconds, then the oracle.jdbc.ReadTimeout should be 604000 milliseconds.
8. In the **Statement Cache Size** field, enter a value from 30 to 40 in a production environment.
9. Click **Save**.
10. Repeat steps 3 to 9 for all other OSM JDBC connections.
11. Click the shopping cart. From the shopping cart, click **Activate Changes**.

Creating and Configuring Persistent File Stores

A persistent file store provides storage for WebLogic Server subsystems and services that require persistence. For example, a file store can store JMS messages. The OSM installer uses the default file store for each managed server and associated OSM JMS servers in the cluster.

Oracle recommends that you create a persistent file store for each managed server in the cluster and for the associated JMS server pairs of each managed server. If the managed servers are configured for whole server migration, ensure that the persistent stores are located in a shared directory.

To create and configure a persistent file store:

1. Create or ensure that a directory exists for the persistent file stores. The directory must exist on your system, so ensure that it is already created.

   > **Note**
   >
   > When a custom file store is targeted to a migratable target, the specified directory must be accessible from all candidate server members in the migratable target. For highest reliability, use a shared storage solution that is itself highly available, for example, a storage area network (SAN) or a dual-ported SCSI disk.

2. Log in to the WebLogic Remote Console.

   The WebLogic Remote Console is displayed.

3. Click **Edit Tree**.
4. Expand **Services** and select **File Stores**.

   The Summary of File Stores page is displayed.

5. Click **New**.

   The Create a New File Store page is displayed.

   Enter the following:

   - **Name**: The name of the file store.
   - **Directory**: The path name to the directory on the file system where the file store is kept.

   > **Note**
   >
   > After you create a file store you cannot rename it. You must delete it and create another one with a different name.

6. Select **Target** tab. Then select the server instance or migratable target on which you want to deploy the file store.
7. Click **Save**.
8. Select the file store you just created.
9. Click the **General** tab. Then select**Show Advanced Fields** and enter the following:

- **Logical Name**: Optional name that can be used by subsystems that need a way to refer to different stores on different servers using the same name.
- **Synchronous Write Policy**: Specifies how this file store writes data to disk.

10. Click **Save**.
11. In the left pane, expand **Services**. From **Services**, select **JMS Servers**.

    The Summary of JMS Servers page is displayed.
12. In an OSM cluster, the OSM installer creates the following JMS servers in pairs:

    oms_jms_server_managed_server_
    osmJmsNonMigratableServer_managed_server_

    where *managed_server* is the managed server that the JMS server is associated with.

    > **Note**
    >
    > For a non-clustered managed server, the name of the JMS server is: **oms_jms_server**.

    Select the one from the managed server pair that will use the custom file store.
13. On the settings page for the JMS server, click the **Configuration** tab, and then the **General** sub-tab.
14. In **Persistent Store**, select the custom file store that you want for the JMS server.
15. Click **Save**.

    For more information about configuring general JMS server properties, see WebLogic Remote Console.
16. In the Remote Console, click the shopping cart. From the shopping cart, click **Commit Changes**.

    > **Note**
    >
    > Not all changes take effect immediately. For some changes you must restart WebLogic Server.
17. Repeat steps 11 to 16 for the other JMS server in the JMS server pair.
18. Repeat steps 3 to 16 for each managed server in the cluster.

Copying Metric Rule Files

A set of XML files called ADML files contains the metric rules that allow the system to collect aggregated metric data about your system. OSM Order Metrics Manager allows the data that the system gathers to be displayed in any metric client. The OSM installation includes an interface called Oracle Dynamic Monitoring Service (DMS), which displays this metric data in a set of metrics tables.

In some configurations, the *Middleware_home* location for the administration server is separate from *Middleware_home* locations for OSM managed servers. In this case, you must remotely

copy the **domain-oracle_comms_osm-11.0.xml** file from the ADML directory of an OSM managed server to the ADML directory of the administration server.

The metric rules files are automatically installed in the correct directory when you first start the OSM server. Metric rules files are installed in the ADML directory, as in the following example:

```
Middleware_home/oracle_common/modules/oracle.dms/adml
```

If this directory is not accessible to the OSM server, you must copy the files using the procedure in this section.

> **Note**
>
> You might have to run the procedure in this section several times. If the administration server does not share the domain directory or the *Middleware_home* directory with a managed server, the system does not create the **oms_dms** directory and you must copy ADML files manually to the administration server's *Middleware_home* directory.

To copy metric rules files:

1. Start an OSM WebLogic managed server.
2. Go to the *domain_home***/osm_dms** directory.
3. As a UNIX or Linux user with write permission to the ADML directory, run the following script:

   ```
   copyAdmlFiles.sh
   ```

   This script copies the ADML files **domain-oracle_comms_osm-11.0.xml** and **server-oracle_comms_osm-11.0.xml** to the *Middleware_home*/**oracle_common/modules/oracle.dms/adml** directory.

For a graphical representation of the metrics data, you can use the Oracle Communications Application Management Pack interface for Oracle Enterprise Manager Cloud Control 12*c*. For more information about installing and configuring Application Management Pack, see *Oracle Application Management Pack for Oracle Communications System Administrator's Guide*. For more information about Enterprise Manager, see *Oracle Enterprise Manager Cloud Control Basic Installation Guide*.

> **Note**
>
> If you are using Internet Explorer 11 as the web browser to access Oracle Enterprise Manager Cloud Control 12.1.0.4, you must update Oracle Enterprise Manager Cloud Control 12.1.0.4 with the following patch:
>
> Patch 20465665: MERGE REQUEST ON TOP OF 11.1.1.7.1 FOR BUGS 18680326 14739854 19933990
>
> Download the patch from the Oracle support website at:
>
> https://support.oracle.com
>
> Instructions for applying the patch are provided in the patch Read me file.

For more information about accessing the DMS interface to view metrics tables, as well as using Application Management Pack to graphically view metrics data, see the topic about metrics data in *OSM System Administrator's Guide*.

Relocating ADML Files Without Restarting the Server

Whether the files are copied automatically by the OSM server or manually, DMS locates the new ADML files when you restart the WebLogic servers. You can also have DMS locate the ADML files immediately, without restarting the servers.

To have DMS locate new ADML files without restarting the servers:

1. If you are a UNIX or Linux user, run the following script:

   ```
   Middleware_home/oracle_common/common/bin/wlst.sh
   ```

   > **Note**
   >
   > You cannot use the **wlst.sh** script in the *WebLogic_home***/common/bin** directory because this script does not have the Oracle JRF extensions enabled.

2. At the command prompts, run the following command:

   ```
   connect('user','password','t3://hostname:port')
   ```

   where *user* is the user name of the WebLogic administrator, *password* is the password of the WebLogic administrator, *hostname* is the name of the administration server and *port* is the administration server port number. For more information about using WLST over a secure connection, see the security section of *OSM System Administrator's Guide.*

3. Load the metric rules files by running the following command:

   ```
   reloadMetricRules()
   ```

   The system loads the metric rules files in the ADML directory. This command shows the metric rules of all the managed servers and the administration server. If one of the OSM ADML files is missing from the reported files list for a server, ensure that the *Middleware_home* directory for that server has the ADML files correctly deployed.

4. Open the **setDomainEnv.sh** file from the *domain_home*/**bin** directory of the administration server.

5. Add the following string to the JAVA_OPTIONS property:

   ```
   -Dweblogic.management.disableManagedServerNotifications=true
   ```

Registering Oracle HTTP Server Instance

To view metrics data using Application Management Pack, you must have Oracle HTTP Server configured and running in OSM. For more information, see *Setting up an Oracle HTTP Server for OSM Cluster Load Balancing* [Doc ID 1618630.1] knowledge article on the Oracle support website at:

https://support.oracle.com

You must also register the Oracle HTTP Server instance in OSM.

To register the Oracle HTTP Server instance:

1. Start an OSM WebLogic administration server.
2. Go to the **bin** directory of the Oracle HTTP Server instance, for example, *OHS_Instance_HOME*/**bin**.
3. Run the following command:

   ```
   opmnctl registerinstance -adminHost hostname -adminPort port
   ```

   where *hostname* is the name of the OSM WebLogic administration host machine and *port* is the port of the OSM WebLogic administration server.

   The Oracle HTTP Server instance is registered in the OSM server.

Queue Configuration Post Installation Tasks

The following sections provide instructions for configuring OSM and solution queues after installing OSM.

Configuring Distributed Queues for an OSM Solution

In addition to the distributed queues that OSM creates during installation, you may need to create and configure your own custom distributed queues to support your particular OSM solution.

Use the following high-level steps to create and configure a distributed queue and its members:

- Create the member queues, one per managed server pinned to each JMS server.
- Create the distributed queue with equally-weighted member queues and target the distributed queue to the cluster.
- Confirm that the distributed queue is accessible and visible to all managed servers in the cluster.

To create the distributed queue and its member queues:

- In WebLogic Server 14.1.2, distributed queues are not directly visible in the Remote Console. However, their individual member destinations are visible and manageable through the console interface. Additionally, the creation of distributed queues is not supported from the console. To configure distributed queues and their members, use the **WebLogic Scripting Tool** (WLST).
- A sample WLST script for creating a distributed queue and its member destinations (`create_weighted_distributed_queue.py`) is available in the SDK at **osm-sdk/SDK/Samples/wlst.** If you only need to create and add members to an existing WDQ then comment out the lines indicated within the inline comment of the script.

To confirm that the distributed queue is created successfully:

- You can use WLST to verify if a distributed queue has been successfully created. A sample script (`get_all_weighted_distributed_queues.py`) that gets all the distributed queues is available in the SDK at **osm-sdk/SDK/Samples/wlst**
- This script retrieves all configured distributed queues along with the names of their associated member queues.
- While distributed queues are not directly visible in the WebLogic 14.1.2 Remote Console, their individual member queues remain visible and more details of member queues can be viewed through the console interface.

Configuring Separate Error Queues

Messages added to an error queue from an internal OSM queue are encoded, so you cannot determine which queue the error originated from. To assist in troubleshooting, you can create separate error queues with appropriate names for each of the internal OSM queues and target the OSM queues to these new error queues. For more information about creating queues, consult the Oracle WebLogic Server documentation.

OSM Integration with External Systems

The following sections describe OSM integration tasks between OSM, Oracle Communications ASAP, and Oracle Communications Unified Inventory Management (UIM). These integration tasks are also applicable for OSM integration with other remote applications.

> **Note**
>
> When you are using **JTA with XA-enabled messaging** with either SAF (Store-and-Forward) messaging or JMS bridge messaging (with *Exactly-once* QoS), it is recommended to enable Cross Domain Trust instead of Global Trust.
>
> Enabling Global Trust introduces security risks, especially for anonymous users and has been observed to cause JMS message rollbacks with errors. To ensure secure and reliable message delivery, always configure Cross Domain Trust.

Configuring Domain Trust

You should enable domain trust when SAF is configured, as it is needed for robust inter-domain communication using distributed destinations. Domain trust relies on the use of a shared password, which provides access to all domains that participate in the trust. Strict password management is critical.

If you use SAF without domain trust configured, you may experience unstable SAF behavior in your environment.

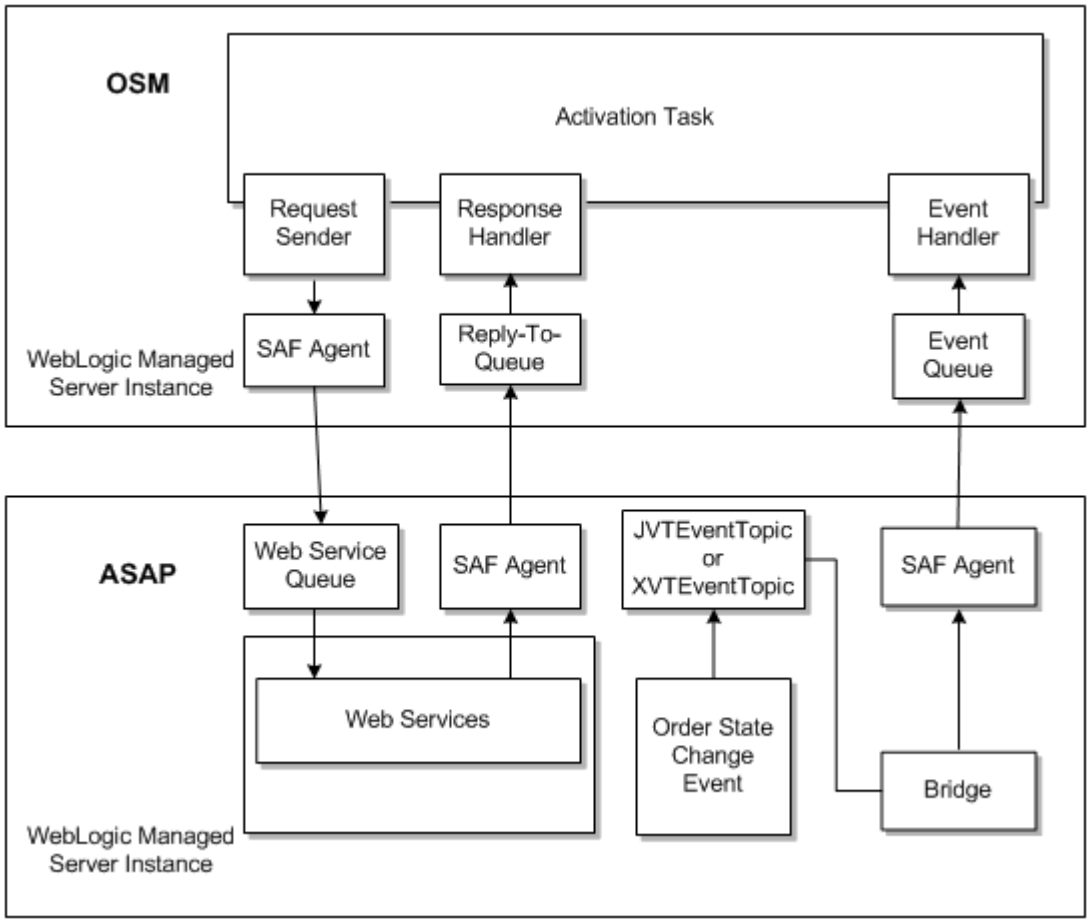
For details about enabling global trust, see "Enabling Global Trust" in *Oracle Fusion Middleware Administering Security for Oracle WebLogic Server*.

Integrating OSM and ASAP or IP Service Activator Using SAF Agent and JMS Bridging

To ensure reliable communication Oracle recommends that you create a Store-and-Forward (SAF) agent and JMS bridge between the Oracle Communications ASAP WebLogic server or Oracle Communications IP Service Activator WebLogic server and the OSM WebLogic server.

Figure 7-1 illustrates an example SAF and JMS bridge configuration between the web service interface on ASAP and a web service client on OSM. An example SAF and JMS bridge configuration between the web service interface on IP Service Activator and a web service client on OSM would appear the same.

Figure 7-1 SAF Agent and JMS Bridge Configuration Between OSM and ASAP



In this example, an OSM SAF agent sends requests to the ASAP request queue, and ASAP returns responses through the ASAP SAF agent to the OSM reply-to queue. In addition, ASAP sends work order state changes from the JSRP XVTEventTopic through a JMS bridge with a SAF agent to the OSM event queue.

For detailed instructions for creating SAF and JMS bridges between ASAP and OSM, see *Configuring WebLogic Resources for OSM Integration With ASAP And UIM On Different Domains (Doc ID 1431235.1)* knowledge article on the Oracle support website at:

https://support.oracle.com

This article is also applicable to IP Service Activator or to any other remote application that uses a WebLogic JMS server to send and receive web service or JMS messages.

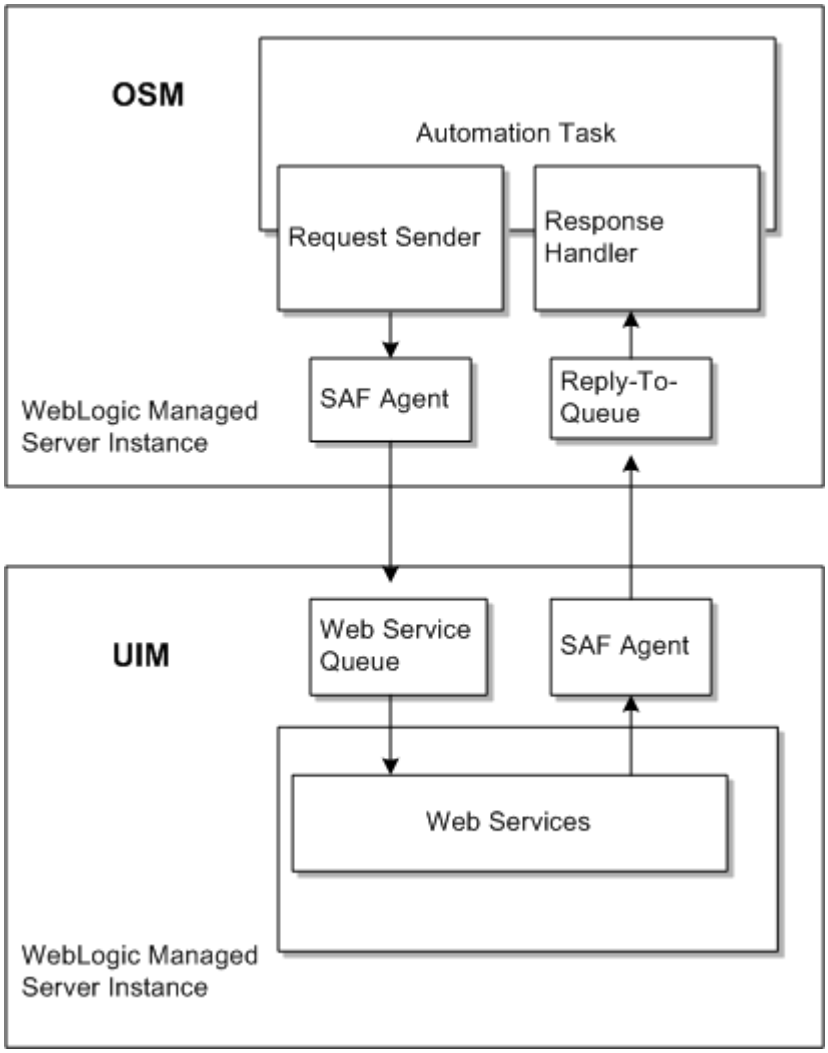
> **Note**
>
> When using the above instructions with any remote application other than ASAP, remember to change any reference to ASAP to the remote application for which you are creating bridges. For example, change ASAP to IPSA.

Integrating OSM and UIM Using SAF Agent

Oracle recommends that you create a SAF agent between the UIM WebLogic server and the OSM WebLogic server. Oracle recommends this SAF agent for the web service interfaces to ensure reliable communication.

Figure 7-2 illustrates an example SAF configuration between the web service interface on UIM and a web service client on OSM.

Figure 7-2 SAF Agent Configuration Between OSM and UIM



In this example, an OSM SAF agent sends requests to the UIM request queue, and UIM returns responses through the UIM SAF agent to the OSM reply-to queue.

For detailed instructions for creating SAF agents between UIM and OSM, see *Configuring WebLogic Resources for OSM Integration With ASAP And UIM On Different Domains (Doc ID 1431235.1)* knowledge article on the Oracle support website at:

https://support.oracle.com

This article is applicable to any remote application that uses a WebLogic JMS server to send and receive web service messages.

Deploying Custom Plug-Ins When Running on Managed Server

If OSM is running on a WebLogic managed server, the **behaviorBuild.properties** file must be configured to specify the URL of the administration server (for example, http://*hostname*:*port*).

From the system prompt, run the command `ant deploy`.

In the WebLogic Remote Console, perform the following steps:

1. Undeploy the custom plug-in from the administration server.
2. Deploy the custom plug-in to the managed server.
3. Target the custom plug-in to the managed server.

Changing the WebLogic Server or Oracle RAC Database Size

The following sections describe how to change the number of managed servers in a cluster or the Oracle RAC database size.

Connecting Oracle RAC with JDBC Multi Data Source

During installation, the OSM installer prompts for the database parameters of the Oracle RAC instances and automatically creates the appropriate configuration in a JDBC multi data source. If you decide to manually configure your JDBC data source, you must understand the following discussion.

At the application layer, OSM maintains WebLogic Server order affinity. That is, all processing of an order is performed exclusively by one WebLogic Server instance, to minimize serialization. The OSM application cannot ensure all database operations for a particular order maintained by a WebLogic Server instance are directed to the same Oracle RAC instance. Thus, the approach to preserve Database Server order affinity is to have each WebLogic Server instance connect to only one Oracle RAC instance at any instant.

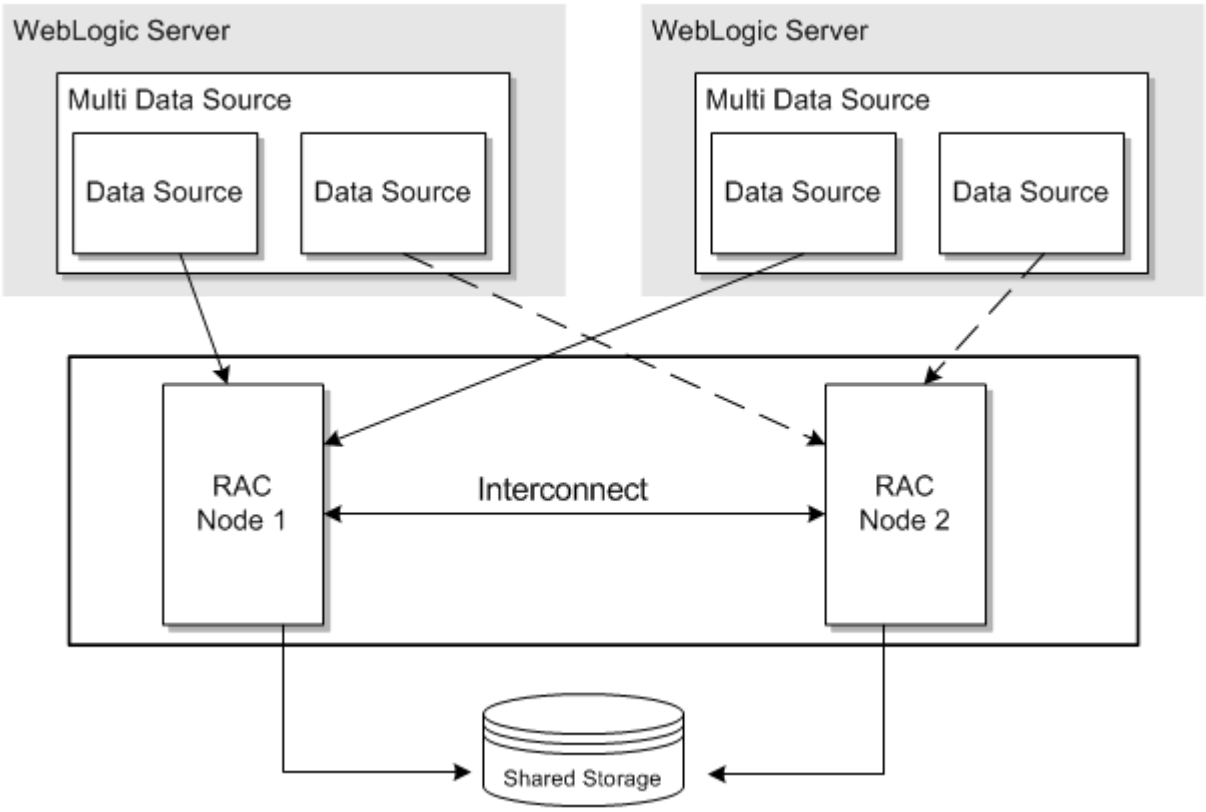
OSM uses a multi data source consisting of two data sources, each of which connects to an Oracle RAC database instance. Using the failover algorithm, the first data source is the primary data source and the other data source is the secondary data source. Under normal operation, only the primary data source is connected and used. When the primary data source fails, the multi data source chooses the next available data source as the primary data source.

The failover algorithm is used in both active-passive and active-active topologies. However, the configuration of the data source members within the multi data source is different:

- In active-passive Oracle RAC, all instances in the WebLogic Server cluster are configured to the PREFERRED Oracle RAC database instance as the primary data source, and to the AVAILABLE Oracle RAC database instance as the secondary. Upon database failure, all WebLogic Server instances transition from the PREFERRED Oracle RAC database instance to the AVAILABLE Oracle RAC database instance.

  Figure 7-3 illustrates this configuration.

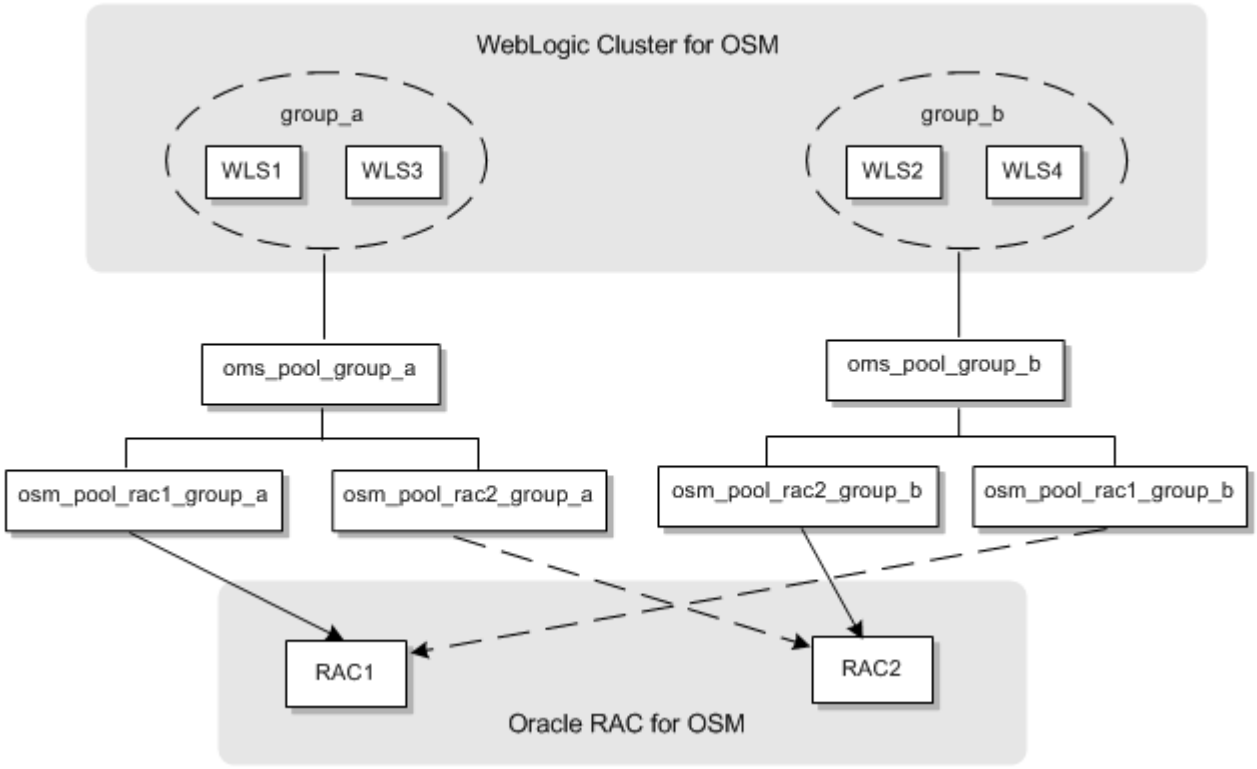
Figure 7-3 Data Source Configuration for Oracle RAC Active-Passive



- In active-active Oracle RAC, WebLogic Server instances are partitioned. Half of the WebLogic Server instances are configured to one Oracle RAC database instance as the first data source and to the other Oracle RAC database instance as the second data source. The other half in the WebLogic Server cluster are configured with the sequence of the Oracle RAC database instances swapped. As a result, if one of the Oracle RAC database instances fails, half the WebLogic Server instances failover to the remaining Oracle RAC database instance, which is already handling the database operations of half of the WebLogic Server cluster.

  Figure 7-4 illustrates this configuration.

Figure 7-4 Data Source Configuration for Oracle RAC Active-Active



In the active-active Oracle RAC configuration, the use of the failover algorithm (as opposed to a load-balancing algorithm) may appear counter-intuitive. However, keep in mind that load balancing in an active-active Oracle RAC configuration is not managed at the multi data source layer, but rather by partitioning the instances in the WebLogic Server cluster. There is no dynamic load balancing between WebLogic instances and database instances.

The relationship of a WebLogic instance and database instance in an active-active Oracle RAC configuration is many-to-one. That is, more than one WebLogic instance may choose the same database instance as its primary database instance, but a WebLogic instance cannot choose more than one database instance as its primary database instance. When you add a new database instance, you can either reassign an existing WebLogic instance to it, or create a new WebLogic instance.

WebLogic Server instances must be partitioned appropriately to load-balance with active-active Oracle RAC. The recommended approach is to have an even number of physical application servers with the same hardware dimensioning and weight. You should monitor the performance of your WebLogic and database instances to ensure they are not overloaded or under utilized. You can add more WebLogic instances to a database instance that is not fully utilized, or reassign a WebLogic instance to another database instance that is overloaded.

When a WebLogic Server cluster resizes, the ownership of an order is reassigned. The cache transfer of records from one database instance to another has a temporary impact on performance.

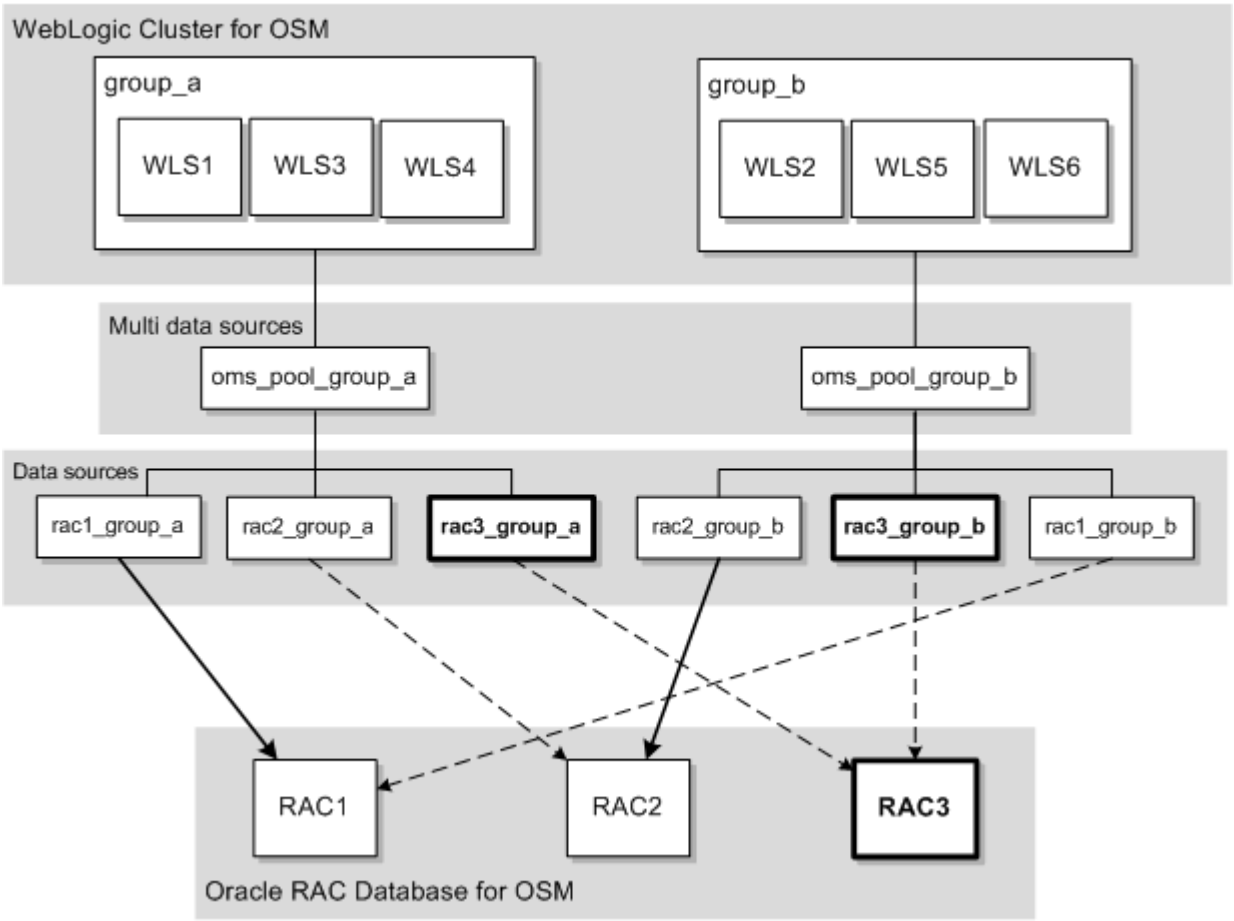
Adding Oracle RAC Instances

You can add a Oracle RAC instance to your environment in the following roles:

- As an additional passive backup instance in an active-passive environment.
- As a passive backup instance in an active-active environment. The new instance connects to the existing WebLogic partitions through new secondary data sources connected to the existing multi data sources.

  Figure 7-5 shows an example of adding a third Oracle RAC instance as a backup in an active-active environment. The figure shows the new Oracle RAC instance and data sources in bold. Note that no WebLogic partition uses RAC3 as a primary instance.

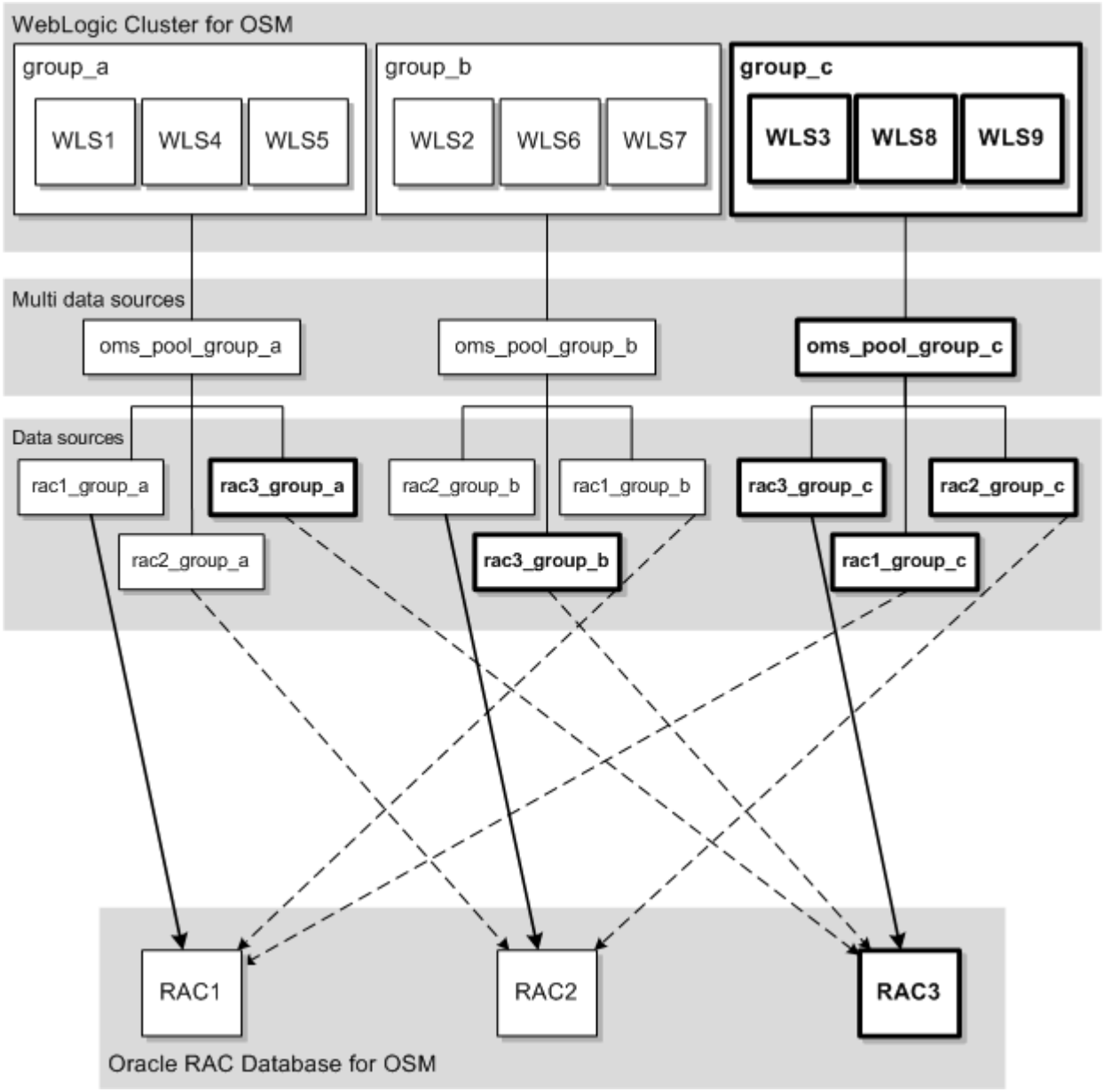
Figure 7-5 Data Sources for a Third Backup Oracle RAC Instance in an Active-Active Environment



- As an active instance in an active-active environment. Choose this option for higher availability. The new instance connects to:
  - A new WebLogic group through a new primary data source configured in a new multi data source.
  - The existing WebLogic groups through new secondary data sources configured in existing multi data sources.

  Figure 7-6 shows an example of adding a third Oracle RAC instance as an active member of an active-active environment. The figure shows the new elements in bold. Note that each Oracle RAC instance serves as a primary instance for a new WebLogic group.

Figure 7-6 Data Sources for a Third Active Oracle RAC Instance in an Active-Active Environment



See the following for more information:

- For grouping and adding new managed servers, see the WebLogic Server documentation and "Adding a New Managed Server to a Clustered Environment."
- For adding a new Oracle RAC instance and partitioning the OSM database schema, see the Oracle Database documentation and *OSM System Administrator's Guide*.
- For adding new data sources and multi data sources, see "Manually Configuring Additional Data Sources for an Oracle RAC Instance."

Manually Configuring Additional Data Sources for an Oracle RAC Instance

OSM supports high availability in the database layer through configuration of Oracle Real Application Clusters (Oracle RAC) for either:

- Failover, also known as warm standby or active-passive Oracle RAC.
- Load balancing, or active-active Oracle RAC.

Each WebLogic Server instance in the OSM cluster interacts with Oracle RAC through a WebLogic multi data source configured for failover with multiple data sources, one for each Oracle RAC instance. This setup is used for both active-passive and active-active configurations. In an active-active configuration, load balancing is achieved by evenly distributing the clustered server nodes into groups in an alternating fashion.

During installation, the OSM installer automatically creates the appropriate data source configuration for the first Oracle RAC instance. You can let it automatically configure data sources for additional Oracle RAC instances, or you can choose to manually configure additional data sources after installation.

If you choose to manually configure additional data sources after installation, the Installer automatically creates the following:

- Two data sources: **osm_pool_rac1_group_a** and **osm_pool_rac1_group_b.**
- Two multi data sources: **oms_pool_group_a** and **oms_pool_group_b**

You must configure additional data sources as described in "Manually Creating and Configuring Data Sources." The number of additional data sources you configure depends on the number of Oracle RAC instances and your environment's configuration type.

For a second Oracle RAC instance:

- In an active-passive configuration, create one new data source.
- In an active-active configuration, create two new data sources.

For additional Oracle RAC instances beyond the second:

- In an active-passive configuration, the new Oracle RAC instance is an additional backup instance. Create one new data source to connect the existing multi data sources to the new Oracle RAC instance.
- In an active-active configuration, the new Oracle RAC instance can be one of the following:
  - A passive backup instance. Create new data sources connecting each existing multi data source to the new Oracle RAC instance. See Figure 7-6 for a detailed example.
  - An active instance. Create a new multi data source and 2*n*-1 new data sources, where *n* is the total number of Oracle RAC instances. Of the new data sources, *n* connect the new multi data source to the Oracle RAC instances, and *n*-1 connect the existing multi data sources to the new Oracle RAC instance.

    For example, if you are adding a third Oracle RAC instance, you would create five new data sources: three to connect the new multi data source to each Oracle RAC instance, and two to connect the existing multi data sources to the new Oracle RAC instance.

    See Figure 7-6 for a detailed example.

Manually Creating and Configuring Data Sources

To create and configure additional data sources for an Oracle RAC instance:

1. Log in to the WebLogic Remote Console.
2. In the **Edit Tree**, expland **Services**. From **Services**, select **Data Sources**.
3. On the Summary of JDBC Data Sources page, click **New**.

4. On the JDBC Data Sources Properties page, do the following:
   - Enter a name for the JDBC data source.
     - If you have an active-passive configuration, name the data source **osm_pool**_*n* where *n* is the number of the new Oracle RAC instance.

       For example, if you are adding a third Oracle RAC instance, name the new data source **osm_pool_3**.
     - If you have an active-active configuration, name the data source **osm_pool_rac***n*_**group**_*x*, where *n* and *x* represent the Oracle RAC instance and WebLogic group that the data source connects.

       For example, if you are adding a third Oracle RAC instance, one of the new data sources will be called **osm_pool_rac3_group_c**. It connects to the third Oracle RAC instance and the WebLogic group called **group_c**.
   - Enter a unique JNDI name for the data source in either configuration. For example, **oracle/communications/osm/internal/jdbc/pool_3**.
   - Select **Oracle** as the database type.
   - Click **Next**.
5. Select one of the following non-XA thin drivers and click **Next**:
   - If your Oracle RAC database is configured with a remote listener (the default), select ***Oracle's Driver (Thin) for RAC Service-Instance connections; Versions: 10 and later.**
   - If your Oracle RAC database is configured with local listeners, select ***Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later.**
6. On the Transaction Options page, do the following:
   a. Ensure the default option **Supports Global Transactions** is selected.
   b. Select the **Logging Last Resource** option.
   c. Click **Next**.
7. On the Connection Properties page, specify the service name, database name, host, and port of the additional Oracle RAC instance based on one of the following scenarios:
   - If your Oracle RAC database is configured with a **remote listener** and server-side load balancing (the default):
     a. Specify the same the host, port, and service name as those specified for the existing data sources.
     b. Specify the unique instance name of the new Oracle RAC instance. The instance name is required in order to override server-side load balancing. See "Listener Considerations for Oracle RAC" for a full discussion of listener functionality.

     For example, if an existing data source URL is:

     ```
     jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
     (HOST=host1.oracle.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=orcl)
     (INSTANCE_NAME=orcl1)))
     ```

     The new data source URL will specify the same host, port, and service name, and a unique instance name, as follows:

     ```
     jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
     (HOST=host1.oracle.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=orcl)
     (INSTANCE_NAME=orcl2)))
     ```

- If your Oracle RAC database is configured with **local listeners** only:
  a. Specify either a different host or port than the host or the port of the existing data sources.
  b. Specify either the same service name or a unique instance name, depending on which the existing data sources specify. If the existing data sources specify both, the new data source must as well.

  For example, if the existing data source URL is:

  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
  (HOST=host1.oracle.com)(PORT=1521)))(CONNECT_DATA=(INSTANCE_NAME=orcl1)))
  ```

  The new data source URL will specify a different host, and only specify the instance name, as follows:

  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)
  (HOST=host2.oracle.com)(PORT=1521)))(CONNECT_DATA=(INSTANCE_NAME=orcl2)))
  ```

- Enter the OSM database schema user name and password, and then confirm the password.
- Click **Next**.

8. On the Test Database Connection page, do one of the following:
   - Review the connection parameters and click **Test Configuration**.

     WebLogic attempts to create a connection from the administration server to the database. Results are displayed at the top of the page. If the test is unsuccessful, you should correct any configuration errors and retry the test.

     Click **Next**.
   - If the JDBC driver you selected is not installed on the administration server, click **Next** to skip this step.
9. On the Select Targets page, do one of the following:
   - If you have an active-passive configuration, target the new data source to the entire cluster.
   - If you have an active-active configuration, target the data source to one group in the cluster.

     For example, in Figure 7-6, the new data sources ending in **group_a** are targeted to the **group_a** servers, the new data sources ending in **group_b** are targeted to the **group_b** servers, and the new data sources ending in **group_c** are targeted to the **group_c** servers.
10. Click **Finish**.

    Your configuration is saved and the data source is deployed to the cluster.
11. Configure the connection properties for the new data source. See "Configuring Connection Pool Properties."
12. Repeat the previous steps for each required new data source.
13. Add the new data sources to multi data sources. See "Adding Data Sources to Multi Data Sources."

Configuring Connection Pool Properties

In addition to the properties you defined in the previous section, you must configure the data source's connection pool properties.

To configure the connection pool properties, do the following:

1. Select the data source and navigate to the Connection Pool tab.
2. In the **Properties** list, enter **oracle.net.CONNECT_TIMEOUT=10000**.
3. In the **Initial Capacity** field, enter **0**.
4. Click **Advanced** options.
5. Select the **Test Connections on Reserve** check box.
6. In the **Test Frequency** field, enter **300**.
7. In the **Test Table Name** field, enter **SQL SELECT 1 FROM DUAL**.
8. In the **Seconds to Trust an Idle Pool Connection**, enter **10**.
9. In the **Shrink Frequency** field, enter **900**.
10. Save your changes.

Adding Data Sources to Multi Data Sources

You can make all data source and multi data source changes in a single edit session. You do not have to activate your changes between steps.

For any number of new Oracle RAC instances, add the new data sources to the existing multi data sources as described in "Adding Data Sources to an Existing Multi Data Source." For Oracle RAC instances beyond the second instance, you must also create a new multi data source, and add some of the new data sources to it, as described in "Creating a New Multi Data Source."

> **Note**
>
> Oracle recommends that you shut down all managed servers before changing the order of the data sources as described in the procedure below because the order affects the name of the Oracle Coherence cluster when OSM starts.

Adding Data Sources to an Existing Multi Data Source

To add new data sources to an existing multi data source:

1. In the **Edit Tree**, expand **Services**. From **Services**, select **Data Sources**.
2. On the Summary of Data Sources page, click a multi data source name.
3. Click the **Configuration Data Sources** tab, and do one of the following:
   - For an active-passive configuration, move the data source that you want to include in the multi data source to the **Chosen** list.
   - For an active-active configuration, add the data sources to the multi data sources in an rotating order so that no data source appears in the same rank twice.

     For example, if you have three multi data sources, the data sources must appear in the order shown in Table 7-1.

Table 7-1 Example Data Source Order

Multi Data Source	Data Source List
oms_pool_group_a	– osm_pool_rac1_group_a – osm_pool_rac3_group_a – osm_pool_rac2_group_a
oms_pool_group_b	– osm_pool_rac2_group_b – osm_pool_rac1_group_b – osm_pool_rac3_group_b
oms_pool_group_c	– osm_pool_rac3_group_c – osm_pool_rac2_group_c – osm_pool_rac1_group_c

See "Adding Oracle RAC Instances" for an illustration of how these data sources and multi data sources connect to Oracle RAC instances and WebLogic partitions.

4. Save your changes.
5. Repeat the previous steps for each multi data source.

Creating a New Multi Data Source

To create a new multi data source and add the new data sources to it:

1. In the **Edit Tree**, expand **Services**. From **Services**, select **Data Sources**.
2. On the Summary of Data Sources page, click **New** and select **Multi Data Source**.
3. On the Configure the Multi Data Source page, enter the following information:
   - **Name**: Enter **oms_pool_group**_*x* where *x* is the next letter in a sequence.

     For example, if your existing multi data sources are **oms_pool_group_a** and **oms_pool_group_b**, name the new multi data source **oms_pool_group_c**.
   - **JNDI Name**: Enter the JNDI path to where this JDBC data source will be bound.
   - **Algorithm Type**: Select **Failover**.
4. On the Select Targets page, do one of the following:
   - If you have an active-passive configuration, target the new multi data source to the entire WebLogic cluster.
   - If you have an active-active configuration, target the multi data source to the same group in the WebLogic cluster that you targeted most of your new data sources to (typically a newly-added WebLogic server group).

     For example, in Figure 7-6, the new multi data source **oms_pool_group_c** is targeted to the same WebLogic server group as the new data sources **rac3_group_c**, **rac1_group_c**, and **rac2_group_c**.
5. On the Select Data Source Type page, select the **Non-XA Driver** option.
6. On the Add Data Sources page, add the data sources in an order that alternates with that of the existing multi data sources. See Table 7-1 for an example with three multi data sources.
7. Click **Finish**.

Adding a New Managed Server to a Clustered Environment

To add a new managed server to a clustered environment:

1. Ensure that the managed servers and the cluster are created. For information on how to add managed servers, see Oracle Fusion Middleware documentation.
2. Ensure that the managed servers have been assigned to the cluster, and that OSM is installed on the cluster.
3. Create a new managed server and do the following:
   a. If a proxy is used in the cluster configuration, add the new managed server's IP address and port number to the proxy server's configuration file. If necessary, restart the proxy server for the changes to take effect. Refer to your proxy server's documentation for details.
   b. If the cluster address of the cluster is set, update it with the new managed server's IP address and port number.
4. If any cartridge queues are present, create them manually using steps provided in the Queue Configuration Post Installation Tasks section for the to-be added managed server.
5. If an Order-to-Activate cartridge is already deployed, run `ant task config_All`.
6. Run the installer scripts to update the environment with all the resources for the to-be added managed server and deploy the OSM apps. All the OSM default resources like JMS server, queues, persistent store and so on, will be created by the installer for the new managed server.
   Run **discovery.sh** to capture the updated properties related to Oracle Real Application Cluster (Oracle RAC), JMS store and so on. For more information about the properties, refer to Specifying Configuration Properties in the Configuration Phase.

   ```
   $ export PASSPHRASE=passphrase
   $ export OSM_INSTALLER_HOME=</path/to/installerHome>

   $ $OSM_INSTALLER_HOME/scripts/discover.sh -n osm_env_name -c $OSM_CFG_HOME
   ```

   Run **configDB.sh**, if there is a change in the Oracle RAC configuration.

   ```
   $ export PASSPHRASE=passphrase #It should be same as used in discover.sh
   $ export OSM_INSTALLER_HOME=</path/to/installerHome>

   $ $OSM_INSTALLER_HOME/scripts/configDB.sh -n osm_env_name -c $OSM_CFG_HOME
   ```

   Run **configDomain.sh**, to update the domain with all the resources for the to-be added managed server and deploy the OSM apps.

   ```
   $ export PASSPHRASE=passphrase #It should be same as used in discover.sh
   $ export OSM_INSTALLER_HOME=</path/to/installerHome>
   $ export FMW_HOME=</path/to/fmw_home>

   $ $OSM_INSTALLER_HOME/scripts/configDomain.sh -n osm_env_name -
   c $OSM_CFG_HOME
   ```

7. Start all the managed servers.

Removing a Managed Server from a Clustered Environment

When you remove a managed server from a clustered environment, you must first delete the related queues, then remove the managed server using the WebLogic Remote Console. See "Preparing to Remove a Managed Server from a Clustered Environment" and "Removing a Managed Server from a WebLogic Cluster" for more information.

Preparing to Remove a Managed Server from a Clustered Environment

For each queue created for the cartridges on OSM, remove its destination on the to-be-deleted managed server and then delete the queues.

Removing a Managed Server from a WebLogic Cluster

To remove an OSM managed server from a WebLogic cluster:

1. Shut down the Oracle database server or servers used by your OSM instance.
2. Log in to the WebLogic Remote Console.

   The WebLogic Remote Console is displayed.
3. Click **Environment**.
4. Click **Servers**.

   The Summary of Servers screen is displayed.
5. In the **Monitoring Tree**, select all the checkboxes.
6. Select all managed servers. Do not select the administration server (followed by "(admin)" in the list).

   > **Note**
   >
   > Note which OSM managed server you want to remove.
7. Click **Shutdown**.
8. Select **Force Shutdown Now**.

   The State changes from RUNNING to SHUTDOWN.
9. Click **Services**.
10. Click **JMS Modules**.

    The JMS Modules screen is displayed.
11. Click **oms_jms_module**.

    The Settings for oms_jms_module screen is displayed.
12. For every OSM or Order-to-Activate distributed queue, do the following:
    a. Disconnect members of a distributed queue. To disconnect members, you must use WLST as it is not supported through WebLogic 14.1.2 Remote Console. A sample WLST script for disconnecting a member (`delete_weighted_distributed_queue_by_name.py`) is provided in the SDK at **osm-sdk/SDK/Samples/wlst**

b. The same script can also be used to delete the distributed queue after its members have been disconnected. To enable this, you must uncomment the relevant lines that are commented in the script.

13. Delete every *queue_managed_server* or *topic_managed_server*, where *queue* and *topic* are the names of the OSM queues and topics and *managed_server* is the managed server you want to remove.
14. If the managed server you are removing is configured with JMS service migration, remove the osmJmsNonMigratableTemplate_*managed_server* template associated to the managed server you want to delete.
15. Click **Subdeployments**.

    The Subdeployments screen is displayed.
16. Select the oms_jms_server_*ManagedServer* you want to delete.

    If the managed server you are removing is configured with JMS service migration, also select the osmJmsNonMigratableServer_*managed_server* subdeployment associated to the managed server you want to delete.

> **Note**
>
> You cannot delete a managed server subdeployment until there are no resources associated with it. If any resources still appear, either delete the resource if it is a queue, or remove the managed server from the member list if it is a distributed queue.

17. Click **Delete**.
18. Click **Services**.
19. Click **JMS Servers**.

    The Summary of JMS Servers screen is displayed.
20. Select the oms_jms_server_*ManagedServer* you want to delete.

    If the managed server you are removing is configured with JMS service migration, also select the osmJmsNonMigratableServer_*managed_server* JMS server associated to the managed server you want to delete.
21. Click **Delete**.
22. Click **Environment**.
23. Click **Servers**.

    The Summary of Servers screen is displayed.
24. Select the managed server you want to delete.
25. Click **Delete**.
26. Click **Environment**.
27. Click **Clusters**.

    The Summary of Clusters screen is displayed.
28. Click the name of the cluster that the managed server you deleted was associated with.
29. In the **Clusters Address** field, remove the IP address and port number for the managed server you deleted.

30. In the **Number of Servers In Cluster Address** field, reduce the number by one.
31. Click **Save**.
32. If a proxy is used in the cluster configuration, delete the managed server's IP address and port number from the proxy server's configuration file. If necessary, restart the proxy server for the changes to take effect. Refer to your proxy server's documentation for details.
33. Save and close the file.

8

Troubleshooting OSM Installation Problems

This chapter describes some of the issues you may encounter during the Oracle Communications Order and Service Management (OSM) installation process and their solutions.

Artifacts Generated by the Installer

The OSM installer generates the following files and artifacts, which can be found in the **$OSM_CONFIG_HOME/configuration/*environment-name*** directory.
If **$OSM_CONFIG_HOME** is not defined, then the directory is **$HOME/.osm/configuration/*environment-name***.

- **configuration.properties** file: This file is generated by the **discover.sh** script and holds the information you provided and the information that the script discovered about the target environment.
- ***Model.yaml.**: These files represent the last attempted domain configuration to the WebLogic installation in this environment. These are generated after running the **configDomain.sh** and **configOSM.sh** scripts.
- **osm_schema_installs** or **installer_schema_upgrades**: This directory contains the following items, which are generated after running the **configDB** and**configOSM** scripts. The directory **installer_schema_upgrades** gets created if migration is from legacy installer schema:
  - **InstallPlan-OMS-CORE.csv** and **InstallPlan-SEMELE-CORE.csv**: These files contain data related to the DB **InstallPlan** actions along with the status and error messages, if any.
  - **AnalysisReport.xml**: This file contains the OSM schema migration analysis report.
  - **staging**: This directory has model files for the OSM schema as well as the Semele-related models, which have to be created or upgraded.
- **osm-wdt-app-archive.zip**: This archive file represents the last attempted deployment of applications to the WebLogic installation in this environment. This is generated after running the **configDomain** and **configOSM** scripts.
- **update_domain_output**: This directory contains files that provide information about servers and resources that need to be restarted. These files get generated after running the **configDomain** and **configOSM** scripts.
- **wdt_logs**: The **updateDomain.log** file is available under this directory. This contains logs related to domain update. This gets generated by the WDT tool after running the **configDomain** and **configOSM** scripts.

Apart from these, the OSM installer also generates the installer logs for the installer scripts run. These can be found under the directory **$HOME/osm-installer-log/**. Here, you can find the log file **osm-installer-log_YYYYMMDD_HHMMSS.log**. Here, **$HOME** is the user home directory.

Coherence Configuration Error: ORA-00001: unique constraint

The following errors can occur in the OSM WebLogic server logs when creating an order:

```
ORA-00001: unique constraint (ORDERMGMT4701.XPKOM_ORDER_FLOW_COORDINATOR) violated
ORA-00001: unique constraint (ORDERMGMT_OSMPRD.XPKOM_ORDER_HEADER) violated
ORA-00001: unique constraint (ORDERMGMT_OSMPRD.XPKOM_HIST$ORDER_INSTANCE) violated
ORA-00001: unique constraint (ORDERMGMT_OSMPRD.XPKOM_ORDER_INSTANCE) violated
```

These errors occur because incorrect or missing Coherence settings cause the nodes in the cluster to be unaware of each other. The servers are unaware that they must generate order IDs that take the other servers into consideration. This problem does not occur if the same server gets all of the createOrder requests. The problem occurs when any other server gets a request and uses the wrong formula to generate the order ID.

For more information about Coherence, see "Configuring Oracle Coherence for an OSM Cluster."

Coherence Not Able to Start in a Firewall Enabled Environment

When a firewall is configured between the servers, the firewall blocks all communication between the nodes of the coherence cluster. The ports used by coherence communication need to be opened to allow coherence traffic to go through. The coherence cluster port as well as each server's local coherence listening port will need to be opened from the firewall. Each server's local coherence listening port need to be defined by you instead of being allocated by coherence.

> **Note**
>
> The port specified for the local coherence listening port must not be the same as the unicast port used by the coherence cluster.

You can specify the port using the `-D` args as given below to configure the server's local coherence listening port:

```
-D coherence.localport=9000
```

For more details, refer to the following Knowledge Management Articles on My Oracle Support:

- *What Are All the Ports Needed to Be Opened for Coherence (Doc ID 1472388.1)*
- *How To Set Up Coherence Cluster With Firewall Configured Between The Hosted Machines? (Doc ID 2423425.1)*

Error About T3 After Initial OSM Startup

The first time you start the OSM server after installation, you may see an exception indicating **T3 file attachment not found**.

If this occurs, restart the server.

Node Manager Does Not Create IP Address for Whole Server Migration

When you start up a managed server that is configured for whole server migration, the managed server fails to start because node manager does not create the floating IP address for the managed server.

If this occurs, ensure that you have selected **Automatic Server Migration Enabled** when you configured the managed server. Node manager does not allocate IP addresses to managed server unless this value is selected. See "Configure Managed Servers for Whole Server Migration" for information about setting this value.

Handling an OSM Database Schema Installation Failure

When the installer fails during an installation, you receive an error message. Before you continue with the installation, you must find and resolve the issue that caused the failure. There are several places where you can look to find information about the issue.

The database installation action plan spreadsheet is a file that contains a summary of all the installation actions that are part of this OSM database schema installation or upgrade. The actions are listed in the order that they are performed. The spreadsheet includes actions that have not yet been completed. To find the action that caused the failure, do the following:

1. Go to the **$OSM_CFG_HOME/configuration/$*osm_env_name*/osm_schema_installs/*YYYY-MM-DD-HHMMSS*/** and look for files **InstallPlan-OMS-CORE.csv** and **InstallPlan-SEMELE-CORE.csv**.
2. Review the status column in these files. The failed action is the first action with a status that is **FAILED**. The **error_message** column of that row contains the reason for the failure.

The installation log file gives a more detailed description of all the installation actions that have been run for this installation. This log file is located in the **$HOME/osm-installer-log/osm-installer-log_*YYYYMMDD_HHMMSS*.log** file. The failed action is typically at the bottom, that is, the last action that was performed.

Once the issue is resolved, you can rerun the same installer script. It will continue from the point it failed. Remember to rerun the same **configOSM.sh** or **configDB.sh** installer scripts that you used earlier.

Also, the following database tables contain information about the database installation:

- **om_$install$plan_actions**: This contains the same information as the database plan action spreadsheet. Compare this table with the spreadsheet in case of a database connection failure.
- **om_$install$plan**: This contains a summary of the installation that has been performed on this OSM database schema.

Database Connection Problems During Installation

If you receive database connection errors, you can try the following options to fix the issue:

1. If you have an issue while running the **discover.sh** script, verify the information that you have provided in the script (by using the **back** command as required). Correct any errors and try again. If the information provided is accurate, verify connectivity to the database from the host running the installer and rectify any issues. Use the **back** and **next** commands to trigger the re-evaluation of database and to retry the database connection.
2. If an issue arises while running the **configDB** or the **configDomain** script, validate that the database information in this environment's **configuration.properties** is accurate. You should also check for connectivity issues or database server availability issues. If **configuration.properties** is not accurate, you need to rerun the discover.sh script using the existing configuration properties and use this to update the database details. Once the issue is rectified, rerun the install script.

This issue is related to latency in the database connection. Acceptable network latency should be between 0.2 and 0.4 msec. Anything higher than 1 msec can substantially reduce OSM performance.

To verify network latency, do the following:

1. Log in to the machine running the OSM server.
2. Run the following command:

   ```
   #ping -s osm_database
   ```

   where *osm_database* is the host name or IP address of the machine running the OSM database server.

   The system responds with lines similar to the following:

   ```
   PING osm_database: 56 data bytes
   64 bytes from osm_database: icmp_seq=0. time=0.389 ms
   64 bytes from osm_database: icmp_seq=1. time=0.357 ms
   ```

   A value for time of less than **0.4** indicates acceptable network latency. A value greater than **1.0** indicates excessive network latency.

JMS Server Connection Problems

After installation, when you restart the server, you may receive an error message from the JMS server connecting to the database. Many retries of the operation occur.

First, check the database connectivity as the database listener or database instance might be down. As a last resort, you may have to re-create the JMS server resource (not recommended) or re-run the OSM installation.

JDBC Errors When First Order Submitted

If you receive JDBC errors when the first order is submitted to OSM, you may need to turn on JDBC logging. Refer to the Oracle WebLogic Server documentation.

No Users or Groups Are Displayed

After OSM installation, you do not see any users or groups on the **Users and Groups** tab in the WebLogic Remote Console. This is because non-dynamic changes have been made, and the WebLogic administration server (and managed server, if applicable) requires a restart.

To resolve this issue:

1. Restart the administration/managed server to clear the condition.

   If the condition does not clear, proceed with the steps below.
2. Log in to the WebLogic Remote Console and select **Environment**.
3. Select the **Security** tab.
4. Select **Advanced**. If necessary, scroll down the page to find **Advanced**.
5. Select the **Allow Security Management Operations if Non-dynamic Changes have been Made** check box.
6. Click **Save**.

7. Navigate to the **Users and Groups** tab.

   Your users and groups are displayed.

OSM and RCU Installers Are Slow to Run Database Tablespace Query

It can take an unusually long time for the OSM Installer and RCU Installer to run a database tablespace query. Purging the Oracle Database recycle bin ensures that the installers can run the database tablespace query more quickly.

To purge the Oracle Database recycle bin system wide:

1. Log in to SQL*Plus as a user with sysdba privileges.
2. Enter the following command:

   ```
   purge dba_recyclebin;
   ```

   The recycle bin is purged system wide.

To purge the Oracle Database recycle bin for a single user:

1. Log in to SQL*Plus as the OSM installer database user.
2. Enter the following command:

   ```
   purge recyclebin;
   ```

   The recycle bin for the database user is purged.

OSM Installer Issues

You may see the following error if you have outstanding Weblogic edit sessions while configuring OSM in the Weblogic domain:

```
WLSDPLY-09015: updateDomain deployment failed: Domain has outstanding edit
session weblogic, deploy cannot proceed and will exit
```

To fix this issue:

1. Ensure that there is no open configuration modification activity on the domain. This could be happening via scripts invoking WLST or similar APIs, via the WebLogic Remote Console or Enterprise Manager, or a similar user interface.
2. Log out of the WebLogic Remote console.
3. Rerun the script to configure the domain.

Command for unpack.jar Fails with a Write Error

If you run the **unpack.jar** command and you receive a write error, you must provide a target application tag (**-app_dir**) while running the command.

For example:

```
./unpack.sh -template=/scratch/oracle/Middleware/user_projects/domains/
osmprak_72251to730_upgddomain_final8may.jar -domain=/scratch/oracle/Middleware/
```

```
user_projects/domains/osmprak_72251to730_upgddomain -app_dir=/scratch/oracle/Middleware/
user_projects/applications/osmprak_72251to730_upgddomain
```

Managed Servers are Unable to form Coherence Cluster

After restarting the managed servers upon successful installation of OSM, you may see the following warning in the managed server log file:

```
<Warning> <com.oracle.coherence> <BEA-000000> <2021-03-22
04:28:03.513/133.795 Oracle Coherence GE 192.0.2.1
<Warning> (thread=Cluster, member=n/a): Delaying formation of a new cluster;
TcpRing failed to connect to senior Member(Id=1, Timestamp=2021-03-22
04:24:17.89, Address=192.0.2.1:7777, MachineId=43781,
Location=site:location.compute.example.com,machine:192.0.2.1,process:9670,memb
er:M1, Role=c1);
if this persists, it is likely the result of a local or remote firewall rule
blocking connections to TCP-ring port 7777>
```

This issue occurs because the required ports are not enabled in the firewall. As a result, OSM managed servers cannot form a Coherence cluster and the load distribution may not happen properly.

> **Note**
>
> This issue occurs mostly in private cloud environments.

To resolve the issue, enable the following ports in the firewall:

- 7
- 17991, 17992, and 17993

  In the **setDomainEnv.sh** file, enable these ports in JAVA_OPTIONS for all the machines as shown below:

  ```
  export JAVA_OPTIONS="${JAVA_OPTIONS} -Dcoherence.localport=17991 -
  Dcoherence.localport.adjust=17993
  ```

Coherence uses these ports for forming the cluster.

For more details, see the "What Are All The Ports Needed To Be Opened For Coherence?" knowledge article (Doc ID: 1472388.1) on My Oracle Support.

9

Verifying the OSM Installation

This chapter describes how to verify that Oracle Communications Order and Service Management (OSM) is installed correctly.

Checking the State of All Installed Components

You can verify that OSM is installed by checking the state of all installed components in the OSM WebLogic Remote console.

To check the state of all installed components:

1. Log in to the Oracle WebLogic Remote console.
2. In the **Monitoring Tree**, expand **Environment**. From **Environment**, select **Servers**.
3. In the right pane of the console, click the managed server on which OSM is installed.
4. In the tabs, select **Deployments** and expand **oms** to verify that all EJBs and modules have been deployed.
5. Verify that both **oms** and **cartridge_management_ws** applications are **active**.
6. Repeat 2 steps 5 to for all other managed servers in the cluster.

Verifying the OSM Clients

You can verify that OSM is installed by logging into the Order Management web client and the Task web client using the OSM administrator user account. The OSM WebLogic server instance must be running before attempting these procedures.

Some functions and screens provided by these clients are not accessible until you have created and deployed a valid OSM cartridge that includes a role configured with permissions to use these functions and screens. After doing this, you must then assign the role to a workgroup using the OSM Order Management web client. For more information, see *OSM Order Management Web Client User's Guide*.

You can use the product cartridges that come with the OSM SDK and the following sample OSM cartridges included with Oracle Communications Service Catalog and Design - Design Studio:

- Provisioning Broadband and Order Change Demo contained in the **bb_ocm_demo** cartridge file.
- Provisioning View Framework Demo contained in the **view_framework_demo** cartridge file.

For more information about installing these sample cartridges, configuring roles, setting permissions, and deploying cartridges, see the Design Studio Help.

To log in to the Task web client:

1. Access the following URL in your web browser:

   `http://host:port/OrderManagement`

Where *host* is the machine where OSM is installed, and *port* is the server's HTTP port number.

If the server has been set up for secure connection, enter https in the URL. Typically, the System Administrator would have the login information.

2. In the **User Name** field, enter the OSM administrator user name you selected when you installed OSM.
3. In the **Password** field, enter the OSM administrator password you selected when you installed OSM.
4. Click **Login**.

> **Note**
>
> The New Order, Worklist, and Query tabs are not accessible for the OSM administrator. The OSM administrator user can also access the Reporting, Notification, and Options tabs.

To log in to the Order Management web client:

1. Access the following URL in your web browser:

   `http://host:port/OrderManagement/orchestration`

   Where *host* is the machine where OSM is installed, and *port* is the server's HTTP port number.

   If the server has been set up for secure connection, enter HTTPS in the URL. Typically, the System Administrator would have the login information.
2. In the **User Name** field, enter the OSM administrator user name you selected when you installed OSM.
3. In the **Password** field, enter the OSM administrator password you selected when you installed OSM.
4. Click **Login**.

> **Note**
>
> You can log into the Order Management web client, but the administrator user cannot use any functions.

Configuring and Verifying HTTPS Connectivity for OSM Client Browsers

To configure and verify HTTPS connectivity for OSM client browsers:

1. Ensure that you have completed the steps described in "Preparing WebLogic Server for an OSM Cluster Installation."
2. Access one of the following URLs in your web browser:

   `https://host:sslport/OrderManagement`
   `https://host:sslport/OrderManagement/orchestration`

where *host* is the machine in which OSM is installed and *sslport* is the server's HTTPS port number. *host* and *sslport* can also be the host name and SSL secure port of a hardware or software load balancer, for example, if you have set up and configured the Oracle HTTP Server software load balancer.

3. Click **Continue to this website (not recommended)**.
4. Click **Certificate Error**.
5. Click **View Certificates**.
6. Select the root certificate authority.
7. Click **Install Certificate**.
8. Click **Next**.
9. Select **Place all certificates in the following store**.
10. Choose **Trusted Root Certification Authorities**.
11. Click **OK**.
12. Click **Yes**.
13. Select the intermediate certificate authority.
14. Click **Install Certificate**.
15. Import the intermediate certificate into **Intermediate Certification Authorities**.
16. Click **OK**.
17. Click **Yes**.
18. Close the web browser.
19. Open the web browser. No certificate errors should appear.

Configuring OSM to Evaluate System Configuration Compliance

The OSM installation includes an OSM compliance tool. Running this tool evaluates your system's compliance against established rules to ensure that the system is optimally configured for the environment.

This section includes the post-installation tasks required for OSM to interact with the compliance tool.

OSM provides a set of python (*.py) scripts that extend the WebLogic Scripting Tool (WLST) command vocabulary, which allows you to use WLST to run the compliance tool. For more information about WLST, see Oracle WebLogic Scripting Tool documentation. For information about using WLST over a secure connection, see the *OSM System Administrator's Guide.*

Manually Installing Compliance Files

The compliance tool relies on WLST extensions in the *WebLogic_home***/common/wlst** directory. When you start the OSM server for the first time, the WLST extension files are automatically installed in the correct directory. If this directory is not accessible to the OSM server, the files must be installed manually, using the procedure in this section. OSM creates the scripts for manual installation in *domain_home***/osm_compliance/scripts**.

> **Note**
>
> The scripts required for manually installing compliance files will only be created when OSM fails to install the files automatically. Otherwise, the scripts are not created in your environment.

To manually install compliance files:

1. Start an OSM WebLogic managed server.
2. Go to the *domain_home***/osm_compliance/scripts** directory.
3. As a UNIX or Linux user with write permission to the WLST directory, run the following script:

   ```
   copyComplianceWLSTScripts.sh
   ```

   The following WLST extension files are copied to the *WebLogic_home***/common/wlst** directory: **compliance.py**, **evaluate.py**, and **snapshot.py**.

Configuring Compliance for an OSM Cluster

You can run the compliance tool on a single instance of OSM or in a clustered environment. If you are running OSM in a clustered environment, you must configure coherence remote management, which allows you to specify one node as the MBean server that manages all other nodes.

You configure coherence remote management by adding and setting system properties for OSM managed servers after you start the Java virtual machine.

To configure coherence remote management:

1. Log in to the WebLogic Remote Console.
2. Click **Edit Tree**.
3. Expand **Environment** and then click **Servers**.

   The Summary of Servers page is displayed.
4. Click the name of the WebLogic server where you want to designate as the MBean server that manages all other nodes.

   The configuration parameters for the server are displayed on a tabbed page.
5. Click the **Advanced** tab. From the **Advanced** tab, slect **Node Manager** sub-tab.
6. In the **Arguments** field, enter the following:

   ```
   -Dcoherence.management=all
   -Dcoherence.management.remote=true
   ```
7. Click **Save**.
8. In the **Domain Structure** tree, expand **Environment** and then click **Servers**.

   The Summary of Servers page is displayed.
9. Click the name of a WebLogic server where you want to designate as the MBean clients of the MBean server.

   The configuration parameters for the server are displayed on a tabbed page.
10. Click the **Server Start** tab.

11. In the **Arguments** field, enter the following:

    ```
    -Dcoherence.management=none
    -Dcoherence.management.remote=true
    ```

12. Repeat steps 8 and 11 for all other managed servers in your system.
13. Restart the managed servers.

Evaluating System Configuration Compliance

The OSM compliance tool captures a snapshot of your system's configuration and evaluates this configuration against established rules, which are based on best practices and guidelines. Using these rules, the compliance tool analyzes the system and produces an evaluation result that allows you to verify that your system is optimally configured for your environment.

Several default compliance tool parameters are specified in the **oms-config.xml** file. Use these parameters to change the directory path where the system stores the following: the compliance snapshot output, the evaluation results, the snapshot files that will be evaluated, and the evaluation rules. For more information, see *OSM System Administrator's Guide*.

The compliance tool is based on JMX technology and captures the configuration snapshot from target environments, such as JMX MBeans for the WebLogic domain, OSM system and database parameters, and coherence cluster for OSM.

For more information about the rules that are provided and adding new rules, see the compliance tool documentation, provided in the SDK at the following location:

OSM_home`/SDK/Compliance/doc/index.html`

Running the Compliance Tool

Run the compliance tool using WebLogic Scripting Tool (WLST) scripts. For more information about WLST, see WebLogic Scripting Tool documentation.

> **Note**
>
> If you are running the compliance tool in a cluster, there are additional factors to consider. For more information, see "Cluster Considerations."

To run the compliance tool:

1. Enter WLST interactive mode and connect to an OSM managed server.
2. Do one of the following:
   - If you want to create the snapshot then evaluate it do the following:
     a. Run the following command, which creates a snapshot of the system configuration:

        ```
        osmSnapshot("snapshot_directory")
        ```

        where *snapshot_directory* is the directory in which you want to put the snapshot of the system configuration. If you leave this blank, the system puts the snapshot in the default directory specified by the **oms-config.xml** file.

     b. Run the following command, which verifies configuration compliance by evaluating the snapshot:

```
osmEvaluate("evaluate_directory")
```

where *evaluate_directory* is the directory in which you want to put the evaluation of the snapshot against the compliance rules. If you leave this blank, the system puts the evaluation in the default directory specified by the **oms-config.xml** file.

- If you want to take a snapshot and immediately evaluate the snapshot do the following:
  a. Run the following command, which creates the compliance results file:

```
osmCompliance("compliance_directory")
```

where *compliance_directory* is the directory in which you want to put the compliance results. If you leave this blank, the system puts the compliance results in the default directory specified by the **oms-config.xml** file.

Cluster Considerations

You can run the compliance tool in a clustered environment. Keep in mind that the osmSnapshot command collects the following types of configuration:

- **common**: The same on every managed server.
- **coherence**: Available only on the managed server that has the coherence MBean server.
- **server**: Specific to the managed server that is collecting the snapshot.

It is typical to run the compliance tool on the managed server that has the coherence MBean server because doing so will evaluate the broadest set of configuration (common, coherence, and a server). To fully check compliance, you must also run the tool on each of the other managed servers.

Evaluating Compliance Results

The compliance tool saves the evaluation results in the directory specified in the WLST command or in the **oms-config.xml** parameter. The evaluation results are saved in two formats: HTML and XML.

> **Note**
>
> The names of the evaluation result files are in the following formats: *yyyymmdd-hhmmss.xml* and *yyyymmdd-hhmmss.html*.

The evaluation results report displays information about the software versions in your environment, and then lists the results. These results are divided into the following tables:

- **Non-Compliant Rules**: The rules with which your environment is not compliant.
- **Compliant Rules**: The rules with which your environment is compliant.

Table 9-1 lists the columns in the HTML evaluation results file and provides a description for the information contained in each column.

Table 9-1 Compliance Tool Evaluation Results File: HTML

Column Name	Description
Index	Specifies rule index in the table.

Table 9-1 (Cont.) Compliance Tool Evaluation Results File: HTML

Column Name	Description
Rule Name	Displays the name of the compliance rule.
Severity	Defines the severity of this rule. Possible values are: • **Error**: Must fix. Your system might be unstable until you correct the problem. • **Warning**: Should fix. Your system will perform better if you correct the problem. • **Information**: May or may not apply to your system. Evaluate and correct as needed.
Description	Specifies a general description of the rule.
Message	If the rule passes, it displays the compliant message. If the rule fails, it displays a non-compliant message and a list of non-compliant objects.
Rationale	Displays the rationale behind a compliance rule.
Reference	Specifies a link to a document that describes the compliance rule in more detail.
Keywords	Displays keywords related to targeted configurations.

Table 9-2 lists the elements in the XML evaluation results file and provides a description for the information stored in each element.

Table 9-2 Compliance Tool Evaluation Results File: XML

Column Name	Description
compliantResult	Contains the evaluation result of a compliance rule. Each compliant result element stores evaluation the result for one rule.
compliantMessage	Appears if a compliance rule passes. The message is not displayed when the rule fails. This message is copied directly from the rule file.
description	Specifies a general description of the rule.
keywordlist	Displays keywords related to targeted configurations.
nonCompliantMessage	Appears if a compliance rule fails. The message is not displayed when the rule passes. This message is copied directly from the rule file.
nonCompliantObjects	Contains a list of objects that failed the compliance rule. Non-compliant objects are not displayed when the rule passes.
nonCompliantObject	Contains an object that failed the compliance rule.
rationale	Displays the rationale behind a compliance rule.
referenceUrl	Specifies the link to a document that describes the compliance rule in more detail.
ruleName	Displays the name of the compliance rule.

Table 9-2 (Cont.) Compliance Tool Evaluation Results File: XML

Column Name	Description
severity	Defines the severity of this rule. Possible values are: • **Error**: Must fix. Your system might be unstable until you correct the problem. • **Warning**: Should fix. Your system will perform better if you correct the problem. • **Information**: May or may not apply to your system. Evaluate and correct as needed.

10

OSM Pre-Production Testing and Tuning

This chapter describes how to run performance tests and tune Oracle Communications Order and Service Management (OSM) before going into production.

OSM Performance Testing and Tuning Overview

Performance testing and tuning is an iterative process with cycles of testing, tuning, analyzing, and retesting. Although many factors impact OSM performance, you can classify them into the following categories:

- **Hardware**: OSM performance is bounded by the limitations of the hardware on which it runs such as when maximum CPU, memory, or other resources are reached. For example, through the performance testing process, you may discover that you need more hardware for another database instance or for additional WebLogic Server managed servers.
- **Software**: Achieving optimal OSM performance depends on proper configuration and tuning of these OSM components and technologies included in the OSM architecture:
  - The operating systems
  - The Oracle GRID infrastructure and associated disk groups and disks
  - The Oracle Real Application Cluster (Oracle RAC) database
  - The Oracle WebLogic Server cluster
  - Java
  - The Java message service
  - Oracle Coherence
  - Shared storage
  - Oracle Communications Design Studio
- **Solution**: OSM cartridges provide the metadata instructions that the OSM server needs to fulfill orders according to business requirements. The level of complexity defined in OSM cartridges impacts order processing performance. For example, the number of tasks in a process or the number of order line items in an incoming order, and the complexity of the incoming order can affect order throughput. To improve solution-related performance, you may need to redesign the solution in Design Studio and redeploy it.

The main goal of performance testing is to determine how many automation threads and how large an order cache are required in a managed server to handle a peak order rate that is sustainable. Although CPU can sometimes cause performance issues, memory is typically the first resource to reach its maximum capacity. You can determine this peak sustainable order rate by monitoring the memory usage of the WebLogic Server during a performance test.

The process for determining this sustainable peak order rate includes the following steps:

1. Install and configure the performance test environment. This includes the hardware, software, and the OSM solution components.
2. Prepare the WebLogic Server connection pool, maximum constraints for work managers, and the JBoss and Coherence cache size and timeout values by setting these values to

very high settings. Setting these values high enables you to find the point at which the memory of the managed server begins to be overloaded.

3. Find the sustainable peak order rate by monitoring the WebLogic Server's Java heap using JConsole. The sustainable peak order rate is determined by ensuring that the live data size (LDS) remains stable at 50% of the old generation tenured heap.
4. When you have determined the sustainable peak order rate, you can review the WebLogic Server log files to determine the number of automation threads that had been in use, the number of orders in the JBoss and Coherence cache, and the average duration these orders required before completing.
5. Configure the WebLogic Server connection pool, maximum constraints for work managers, and the JBoss and Coherence cache size and timeouts values by setting these values to the those you determined during performance testing. Performing this step ensures that WebLogic Server managed servers in your WebLogic Server cluster do not run out of memory.
6. Use the sustainable peak order rate to determine how many managed servers you require in your WebLogic Server cluster.

Guidelines for the Performance Test Environments

You determined the initial sizing of your production environment hardware when you planned the physical architecture of your system, as described in "Planning the Physical Architecture". In this chapter, you will run performance tests to determine if your initial sizing is adequate.

Ideally, the hardware sizing for the performance test environment should be comparable to that of the production environment. For example, if the test environment is less than half of the capacity of the production environment, then you cannot adequately test the performance capability of the solution. In addition, the technology stack and the solution architecture should resemble the production environment as closely as possible. For example, the Oracle Grid infrastructure, the Oracle RAC setup, the WebLogic Server setup, and the shared storage used should be similar to that used in the production environment.

When the performance test environment is smaller than the production environment, a conservative approach must be taken to extrapolate the results, considering that the results in production may be substantially different than what is being observed in the test environment. For example, you can test a smaller number of managed servers in the test cluster environment as long as the managed servers have the same resources as an equivalent production managed server (for example, memory, CPU). If you plan to deploy multiple managed servers per machine in production, a you should use a similar deployment for testing (for example, smaller number of machines but comparable deployment and resource usage for each machine).

You should not assume a simple, linear extrapolation based on hardware. Often, usage and data contention bottlenecks do not manifest themselves until the system is large enough. Conversely, an undersized system may magnify issues that otherwise would not exist: for instance, when using an Oracle RAC database with a slow interconnect or slow storage retrieval. For example, you must know the number of managed servers you have in your cluster to properly size the coherence invocation service threads. If your coherence threads are not properly sized for your environment, you may experience costly issues in a fully sized production environment.

The performance test environment may also serve as a preproduction environment for tasks such as validating upgrade plans or new cartridges. Oracle recommends that you keep the performance test environment available both prior to the initial deployment and throughout the lifespan of the OSM solution, so that performance testing can be conducted on any new

enhancements, fixes, and workarounds, and any other changes introduced to the implementation.

When you are planning the performance test environment, create a test plan that includes the following high-level information:

- Versions of the following software: OSM, WebLogic Server, Coherence, JDK, Oracle Database, Design Studio, and the OSM Design Studio plug-ins
- The patches applied
- Operating system, version, and configuration
- Solution and deployment architecture
- Latest cartridges
- WebLogic Server configuration files (all files in the *domain_home***/config** directory), JVM heap size, the **osm-coherence-cache-config.xml** file, and the OSM **oms-config.xml** file
- Cluster size
- OSM database configuration: memory size, tablespaces and redo log sizes, layout, and so on. Also, for example, whether you plan to use Oracle RAC and partitioning and, if so, the number of orders per partition.

About Configuring the Environment for Performance Testing

The performance testing process involves determining the sustainable order rate that each managed server in the OSM WebLogic cluster can handle. The sustainable order rate is typically 80% of the maximum order rate that each managed server can handle beyond which the managed server becomes overloaded. This creates a buffer that ensures that large spikes in customer orders do not create problems.

This chapter provides instructions for configuring and tuning work managers, work manager constraints, and JBoss and Coherence caches such that in the initial environment setup, the constraints and caches are set very high. Having these settings high enables you to overload the managed server's cpu and memory so that you can determine the sustainable order rate for each managed server. After determining the sustainable order rate, you can then configure the work manager constraints and caches to ensure that the managed server in the cluster can never process more orders than it can handle.

About Work Managers, Work Manager Constraints, and the JDBC Connection Pool

OSM uses work managers to prioritize and control work. You can tune work managers using work manager constraints, which is an effective way to prevent overloading the managed servers in the OSM WebLogic Server cluster. The work manager constraints limit the number of threads available to OSM work managers.

The OSM installer creates only one maximum thread constraint shared by all OSM work managers. While this performs well in a development environment, this configuration is not the best approach in a production environment. For example, in a production environment under high load, this configuration can cause all available threads to alternate between the automation work manager (osmAutomationWorkManager) and the JMS web service work manager (osmWsJmsWorkManager), impacting core order processing capabilities.

> **Note**
>
> See the *OSM default Work Managers, Constraints related to upgrade process (Doc ID 3019290.1)* knowledge article on My Oracle Support, if you are upgrading from OSM version 7.3.1 or lower (source version) to OSM version 7.3.5 or higher (target version).

To better control the flow of orders, Oracle recommends that you set the following values for thread constraints:

- **osmJmsApiMaxThreadConstraint** for the **osmWsJmsWorkManager** work manager. The osmJmsApiMaxThreadConstraint should be 12.5% of the total number of threads when you initially begin the tuning process.
- **osmHttpApiMaxThreadConstraint** for the **osmWsHttpWorkManager** and **osmXmlWorkManager** work manager. The osmHttpApiMaxThreadConstraint should be 12.5% of the total number of threads when you initially begin the tuning process.
- **osmGuiMaxThreadConstraint** for the **osmTaskClientWorkManager**, **osmOmClientWorkManager** work managers. The osmGuiMaxThreadConstraint should be 25% of the total number of threads when you initially begin the tuning process.
- **osmAutomationMaxThreadConstraint** for the **osmAutomationWorkManager** work manager. The osmAutomationMaxThreadConstraint should be 50% of the total number of threads when you initially begin the tuning process.

After you have completed the tuning process, you will set these constraints to values that enable maximum performance while ensuring that the server does not get overloaded or encounter alternating thread issues.

You must also ensure that every OSM thread always has access to a database connection. Oracle recommends that you set the maximum number of work manager threads to 80% of the database connection pool. One approach to this configuration is to first determine the total number of threads needed by adding all the maximum work manager constraints together, and then set the database connection pool to 125% of this number.

About the JBoss and Coherence Order Cache

OSM uses JBoss and Coherence order caches that determine how many orders can stay in active memory and for how long before being removed from the cache. Tuning the JBoss and Coherence order caches also prevents the managed servers in the OSM WebLogic Server cluster from being overloaded.

Synchronizing Time Across Servers

It is important that you synchronize the date and time across all machines that are involved in testing, including client test drivers. In production environments, Oracle recommends that you do this using Network Time Protocol (NTP) rather than manual synchronization. Synchronization is important in capturing accurate run-time statistics.

Determining Database Size

The size of the database (amount of memory, number of CPUs, and storage capacity) has an impact on performance. Oracle recommends that you run tests using the same size database that is planned for the production environment. You can do this by seeding the database with data, migrating data, or running a representative set of sample orders. Initial testing against an empty database only highlight the most serious problems. With an empty schema, database

performance problems that relate to gathering optimizer statistics that are inaccurate will not become apparent until after the OSM system enters production.

This chapter provides instructions for populating the database with orders, warming up the system, and running database optimizer statistics so that the performance test generates accurate results.

> **Note**
>
> Oracle recommends that you back up the OSM schema before running performance testing. After testing, you can restore the schema so that you do not need to purge orders that were generated during the testing. Keep in mind that exporting and importing the OSM schema can be time-consuming.
>
> Alternatively, you can drop the all partitions and rebuild the seed data. This method can be faster than backing up and restoring the schema.

Setting Up Emulators

If the entire system is not available for testing, you can set up emulators to simulate external requests and responses. For example, if you need to test OSM performance before the billing or inventory system is ready, you can use emulators.

If you are using the Order-to-Activate cartridges, OSM provides an Oracle Application Integration Architecture (Oracle AIA) Emulator, which you can use to emulate order responses. If possible, run emulators on separate hardware from the OSM server so they do not consume OSM resources during performance testing. For more information about setting up and using Oracle AIA emulators, see *OSM Cartridge Guide for Oracle Application Integration Architecture*.

Setting Up a Test Client for Load Generation

You can use a test client to submit the orders for performance testing. The test client can be a custom application or any third-party tool, such as JMeter, LoadUI, or SoapUI. The examples used in this chapter are from a SoapUI project.

Keep the following in mind when setting up a test client:

- Ensure the test client does not impact OSM hardware resources. It is best to run test clients on different hardware from the hardware where OSM is deployed.
- Ensure the number of test client threads is configurable and supported in the test client machine. This is essential for load and scalability testing because the number of concurrent users that the system can support is based on the number of test client threads. If high loading is required, you might need to use multiple test client machines to generate a sufficiently high load.
- Ensure the test client can provide vital statistics on performance data, such as average, maximum, standard deviation, and 90th percentile performance numbers.
- Ensure the test client can complete long running tests with a steady load.

Example Managed Server Configuration

This section describes an example managed server configuration to illustrate how to run a performance test on an OSM instance as described in "Example Performance Tests on OSM Managed Servers".

The example is based on the following:

- Each machine has 64 threads and processors.
- Each machine has 128 GB of physical memory.
- Each machine runs two managed servers.
- Each managed server is configured with 32 GB of memory.

To configure the managed servers, do the following:

1. For each managed server, set 24 hardware threads for garbage collection by using the **-XX:ParallelGCThreads** managed server startup argument.

   See "Configuring Managed Server Startup Parameters" for more information about setting startup parameters.

   > **Note**
   >
   > Even though you could allocate half of the 64 hardware threads for garbage collection, this would create too many tenured heap partitions which increases the risk of fragmentation.

2. For each managed server, configure the database connection pool with four times the number of available hardware threads. This very large connection pool removes any limitations on the number of threads that managed server can use during the performance test. This enables the performance test to determine the actual overload point for the managed servers.

   In the sample environment, given that there are two managed servers on one machine that has 64 threads, the connection pool for each managed server could start at 4 x 64 threads / 2 managed servers = 128 connections for each managed server.

   > **Note**
   >
   > Ideally, each managed server would have its own dedicated machine. In which case, there would be 4 x 64 threads / 1 managed server = 256 connections for the managed server.

   To set the maximum capacity of the connection pool, do the following:

   a. Log in to the WebLogic Remote Console.
   b. In **Edit Tree**, expand **Services**. From **Services**, select **Data Sources**.

      The Summary of JDBC Data Sources screen is displayed.

   c. For each entry in the table with a name in the format:

      osm_pool_*sid*_**group**_*y*

where *sid* is the system identifier (SID) of the Oracle RAC database instance and *y* is the group letter for the managed server, do the following:

Select the data source. The Settings for *pool_name* window is displayed.

Select the **Connections Pool** tab.

In the **Maximum Capacity** field, verify that the value is set to **128**. If it is not, change the value, and click **Save**.

d. Click the shopping cart and click **Commit Changes**.

3. Verify the maximum thread constraints and that the thread constraints are to a large value for initial performance testing. These maximum constraints are calculated based on 80% of the database connection pool, which is 128 x 0.80 = 102:

   - **osmJmsApiMaxThreadConstraint** for the **osmWsJmsWorkManager** work manager. In this example, the constraint should be 102 x 0.125 = 12.75
   - **osmHttpApiMaxThreadConstraint** for the **osmWsHttpWorkManager** and **osmXmlWorkManager** work managers. In this example, the constraint should be 102 x 0.125 = 12.75
   - **osmGuiMaxThreadConstraint** for the **osmTaskClientWorkManager** and **osmOmClientWorkManager** work managers. In this example, the constraint should be 102 x 0.25 = 25.
   - **osmAutomationMaxThreadConstraint** for the **osmAutomationWorkManager** work manager. In this example, the constraint should be 102 x 0.5 = 51.

   To validate these constraints, do the following:

   a. In the Remote Console, select **Edit Tree**. From **Edit Tree**, expand **Environment** and then select **Scheduling**. From **Scheduling**, select **Min Thread Constraints**.

      The Summary of Work Managers screen is displayed.

   b. For each of the constraints listed above:

      Click the name of the constraint. The Settings for *constraint_name* window is displayed.

      Verify that the value of the **Count** field is the value expected from your calculations. If it is not, change the value, and click **Save**.

   c. Click the shopping cart and click **Commit Changes**, if any changes.

4. Verify the size of the **osmGuiMinThreadConstraint** constraint, which is used for the following OSM client work managers:

   - **osmOmClientWorkManager**
   - **osmTaskClientWorkManager**

   To validate the minimum thread constraint for the OSM clients, do the following:

   a. In the Remote Console, in **Edit Tree**, expand **Environment**, and then select **Work Managers**.

      The Summary of Work Managers screen is displayed.

   b. Click **osmGuiMinThreadConstraint**.

      The Settings for *constraint_name* window is displayed.

   c. Verify that the value of the **Count** field is **4**. If it is not then change the value, and click **Save**.

   d. Go to the shopping cart and click **Commit Changes** if you've changed the value.

5. Restart all the managed servers in the cluster.

Guidelines for Performance Testing and Tuning

This section provides general guidelines and example procedures for conducting performance tests on OSM. The sample procedures use SoapUI and JConsole. The method involves testing and tuning the OSM system until you determine the sustainable order rate for each managed server in the OSM WebLogic Server cluster and for the Oracle RAC database instances.

Your performance goals, including the expected results for each test run, are typically based on business objectives.

In working toward achieving the optimum performance for your OSM system, keep the following high-level goals in mind:

- Maximizing the rate by which orders are processed by the system (order throughput)
- Minimizing the time it takes for each order to complete

The performance testing and tuning examples provided in this section illustrate how you can find the correct balance between these considerations, which often affect one another. For example, when you achieve maximum throughput, orders might not have the fastest completion time. Sometimes you must configure OSM to respond faster but at the expense of order throughput.

The example performance test also addresses the technical safety boundaries for the system, such as a hardware resource utility (for example, heap size), and the ability to process a certain number of orders and manual users at the same time. The example performance test also addresses secondary technical goals that ensure the system can continue running under stress, such as handling a large burst of orders, outages of other systems, or failover and recovery of OSM hardware.

> **Note**
>
> Even if you cannot define specific performance requirements, it is valuable to conduct performance testing in order to get benchmark numbers that you can compare with future releases of the solution.

General Guidelines for Running Tests and Analyzing Test Performance

To analyze the results of the test runs, do the following during testing:

- Gather operating system information for OSM application machines.
- Gather Automatic Workload Repository (AWR) snapshots for Oracle RAC, ADDM, and (Active Session History) ASH reports from the database for the exact duration of the test.
- Monitor information for WebLogic Server CPU, heap, and threads using tools like VisualVM or JConsole.
- Gather garbage collection logs and server logs.
- Gather multiple thread dumps regularly, especially during issues.
- Gather heap dumps, if necessary.
- Monitor WebLogic server activities, such as JMS queues, JDBC connections, execute thread pool, and so on, using WLST.

- Monitor network and storage for less latency and consistent throughput based on the documented service times for the hardware.

Example Performance Tests on OSM Managed Servers

When you have created a production ready OSM solution and have deployed it in a test environment as described in "Guidelines for the Performance Test Environments", you can run the performance tests described in this section. The performance testing process includes the following steps:

1. Determine how long your orders last and set the order volatility level on the OSM schema.
2. Begin a performance test to warm up the production system to achieve the following:
   - Enable the OSM server to compile all Java classes involved in processing orders.
   - Enable incremental statistics gathering on low, medium, and high volatility order and gather statistics at appropriate times.
   - Determine the appropriate size of the JBoss and Coherence order cache and the order cache inactivity timeouts.
3. Run the performance test.
4. Gather data.
5. Analyze data.
6. Tune the work manager constraints and the maximum connection pool capacity.

> **Note**
>
> See the *OSM default Work Managers, Constraints related to upgrade process (Doc ID 3019290.1)* knowledge article on My Oracle Support, if you are upgrading from OSM versions 7.3.1 or lower (source version) to OSM versions 7.3.5 or higher (target version).

7. Tune the JBoss and Coherence maximum order cache.
8. Tune the redo log file size.

In addition to the steps described in this chapter, you must also tune other components such as the database, the operating system, the network, the storage, and so on.

Setting the Order Volatility Level

Before you warm up your system, you can specify whether you have low, medium, or high volatility orders for a specific group of database tables. Order with high volatility last for only a few second. Orders with low volatility can last for hours or even days. Orders that are a mix of high or low volatility can be classified as medium volatility. You must specify the order volatility level for these tables because the volatility level is dependent on the solution. For more information about statistics, see *OSM System Administrator's Guide*.

Log in to the OSM core schema and run the following commands to set the order volatility level:

```
execute
om_db_stats_pkg.set_table_volatility('OM_ORDER_FLOW',om_const_pkg.v_volatility_volatility
_level);
execute
om_db_stats_pkg.set_table_volatility('OM_AUTOMATION_CTX',om_const_pkg.v_volatility_volati
```

```
lity_level);
execute
om_db_stats_pkg.set_table_volatility('OM_AUTOMATION_CORRELATION',om_const_pkg.v_volatilit
y_volatility_level);
execute
om_db_stats_pkg.set_table_volatility('OM_ORDER_POS_INPUT',om_const_pkg.v_volatility_volat
ility_level);
execute
om_db_stats_pkg.set_table_volatility('OM_UNDO_BRANCH_ROOT',om_const_pkg.v_volatility_vola
tility_level);
execute
om_db_stats_pkg.set_table_volatility('OM_ORCH_DEPENDENCY_PENDING',om_const_pkg.v_volatili
ty_volatility_level);
```

where *volatility_level* is low, medium, or high.

Warming Up the OSM System

Before an OSM performance test, you must start and run the WebLogic servers and the database so that all Java classes compile and the cache of the database populates with data. Typically, running the system for 5 to 10 minutes at 30 percent of its maximum order intake is enough, at which point the WebLogic server CPU usage and database input/output have stabilized.

After the initial warm-up period, you must run orders again at a higher rate to gather database statistics about low, medium, and high volatility tables. Do not gather statistics unless you have properly warmed up the system or the statistics will not be representative.

While you are running the orders, you must also set various JBoss and Coherence order cache values in preparation for the performance test.

For example, to run a warm up session with SoapUI, do the following:

1. Download and install SoapUI.
2. Create a SoapUI project.
3. Within the project, create a test suite.
4. Create a test step for CreateOrderBySpecification using a single order.
5. Open the order.
6. Click the address bar and select **Add new endpoint**.

   The Add new endpoint screen is displayed.
7. Enter the following:

   `http://hostname:port/OrderManagement/wsapi`

   where *hostname* is the managed server host name or IP address and *port* is the port number of the managed server.
8. Submit the order to verify connectivity with the OSM managed server.
9. Create a load test with the representative set of orders.
10. Open the load test.
11. In the **Limit** field enter a number. For example 600. This value in seconds causes the load test to run for 10 minutes which is enough of the initial warm up time to enable the Java classes to compile.
12. In the **Threads** field, enter **4**.

13. In the **Delay** field, enter a number. For example **15000**. This value represents a 15000 millisecond delay interval (2.5 seconds) in which SoapUI submits the orders. The higher the value, the longer the delay, and the fewer orders are submitted. The lower the value, the shorter the delay, and more orders are submitted.
14. Open a terminal and run the following command:

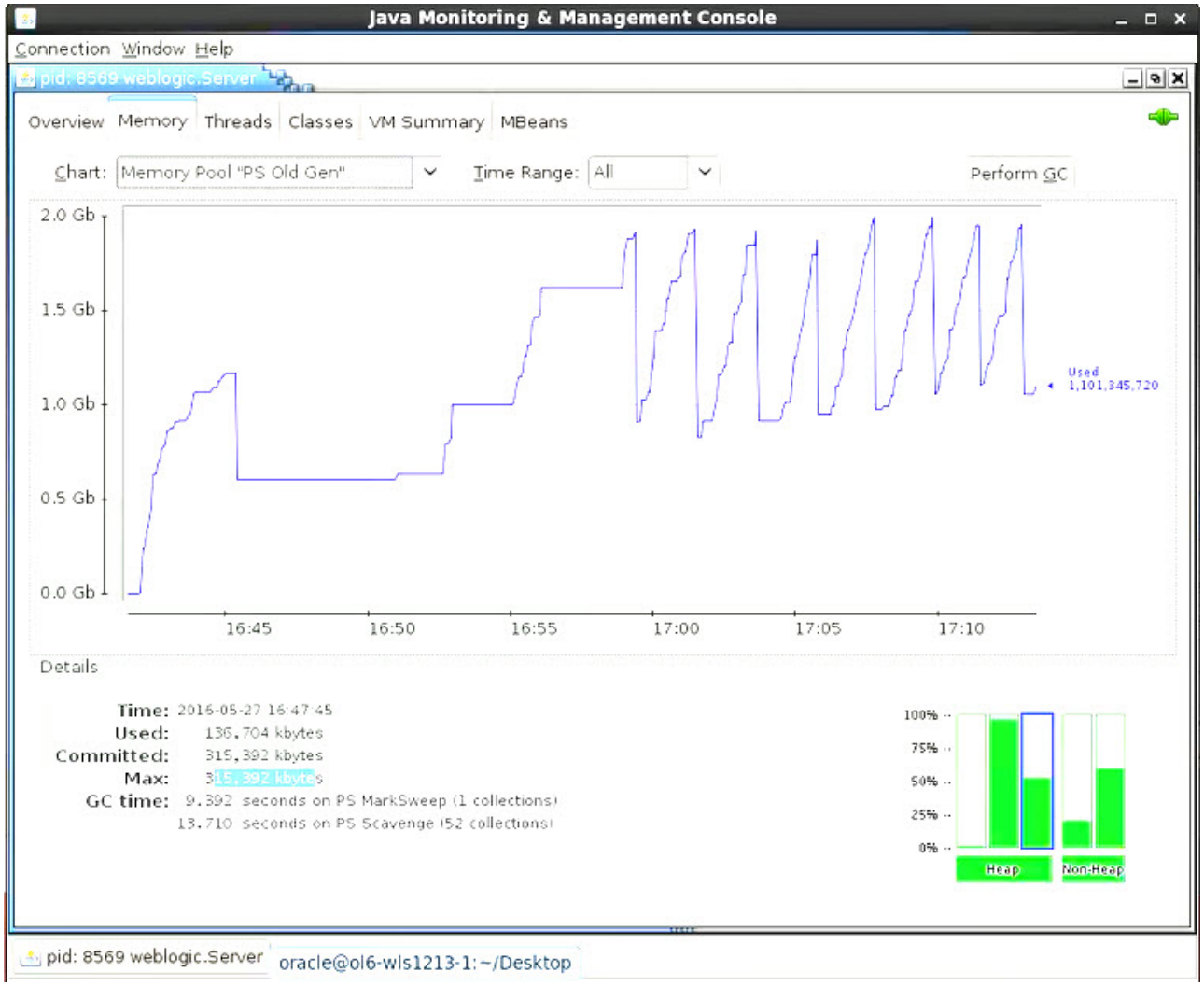
    ```
    ps -ef | grep managed_server
    ```

    where *managed_server* is the name of the managed server you are tuning. The output specifies the current user name running the managed server and the first set of numbers specifies the process ID (PID) number of the managed server.
15. Run *Java_home*/**bin/jconsole** (where Java_home is the JDK root directory) to run the JConsole application for monitoring

    The New Connection screen is displayed.
16. Select the PID that corresponds to the one from the results in step 14.
17. Click **Connect**.
18. Click **Insecure**.
19. Click the **Memory** tab.
20. From the **Chart** list, select **Memory Pool "PS Old Gen"**.
21. Monitor the life data size (LDS) which is the number of live objects that remain after a garbage collection. Ensure that the LDS is no more than 30% of the maximum old generation during the warm up process. If the order rate causes the LDS to increase above 30%, then the managed server is processing too many orders. Increase the **Delay** field amount on the SoapUI Load Test screen until you see the live objects after garbage collection return to 30%.

    For example, Figure 10-1 shows a small 2.5 GB managed server running with more than 30% live objects after garbage collection.

Figure 10-1 JConsole LDS Size Above 30%



22. After the initial performance test to compile the Java classes completes, set the **Limit** value to a higher number. For example **1800** for a 30 minute performance test for gathering database statistics.

> **Note**
>
> The duration you set depends on the OSM system being tested. Some systems may require longer than 30 minutes.

23. Start the performance test for gathering low volatility statistics and determining order cache eviction time.
24. Log in to the order management schema.
25. Enable incremental statistics on low volatility orders.

```
execute om_db_stats_pkg.set_table_prefs_incremental(a_incremental =>
true,a_volatility => om_const_pkg.v_volatility_low);
```

26. After the performance test has completed, gather statistics on the low volatility orders. Because the database is no longer processing orders, you can use all available threads (default behavior) to gather statistics.

    ```
    execute om_db_stats_pkg.gather_order_stats(a_force => true,a_volatility =>
    om_const_pkg.v_volatility_low);
    ```

27. Use the OSM Task web client and verify the average time it takes for orders to complete.

    To verify the average time it take for order to complete, do the following:

    a. Log in to the OSM Task web client.

    b. Click **Reporting**.

    c. Click **Completed Order Statistics**.

    d. In the **From** field, enter the starting time and date for the current performance test.

    e. In the **To** field, enter the current time and date for the performance test.

    f. Click **Find**.

    g. Compare the average time it take for orders to complete from the **Avg Time** column with the longest time it take for orders to complete from the **Highest Time** column. You can use these values to determine how long orders should stay in the cache before being evicted. For example, if orders, on average, take three minutes to complete and it took five minutes for the longest, then a four minute cache eviction timeout for inactivity would be reasonable. Or, if most orders take 30 minutes to complete and the longest took 50 minutes to complete, then a forty minute timeout for inactivity would be enough. The inactivity timeout should capture 80% of your order volume.

28. Using a text editor, open the *domain_home*/**oms-config.xml** file.

29. Add the following text to the bottom of the file before the final </oms-configuration> tag to configure the JBoss cache:

    ```
    <oms-parameter>
        <oms-parameter-name>ClosedOrderCacheMaxEntries</oms-parameter-name>
        <oms-parameter-value>60</oms-parameter-value>
    </oms-parameter>
    <oms-parameter>
        <oms-parameter-name>ClosedOrderCacheTimeout</oms-parameter-name>
        <oms-parameter-value>60</oms-parameter-value>
    </oms-parameter>
    <oms-parameter>
        <oms-parameter-name>OrderCacheMaxEntries</oms-parameter-name>
        <oms-parameter-value>order_max</oms-parameter-value>
    </oms-parameter>
    <oms-parameter>
        <oms-parameter-name>OrderCacheInactivityTimeout</oms-parameter-name>
        <oms-parameter-value>inactivity_timeout</oms-parameter-value>
    </oms-parameter>
    ```

    where

    - *order_max* is the maximum number of orders that can be in the managed server's JBoss cache. Set this value to a high number, such as 2000 for the purposes of performance tuning procedure. You will change this number to a lower setting after completing the performance tuning procedure.
    - *inactivity_timeout* is the value you determined in step 27 in seconds. This timeout evicts an order from the JBoss cache.

30. Save and close the file.

31. Using a text editor, open the **osm-coherence-cache-config.xml** file.
32. Search on **osm-local-large-object-expiry**:

```
<scheme-name>osm-local-large-object-expiry</scheme-name>
     <eviction-policy>LRU</eviction-policy>
     <high-units>order_max</high-units>
   <low-units>0</low-units>
     <unit-calculator>FIXED</unit-calculator>
        <expiry-delay>inactivity_timeout</expiry-delay>
        <flush-delay>1000ms</flush-delay>
  </local-scheme>
```

    where

    - *order_max* is the maximum number of orders that can be in the managed server's Coherence cache. Set this value to a high number, such as 2000 for the purposes of performance tuning procedure. You will change this number to a lower setting after completing the performance tuning procedure.
    - *inactivity_timeout* is the value you determined in step 27 in seconds. This timeout evicts an order from the Coherence cache.

33. Save and close the file.
34. Restart all servers.
35. In the SoapUI load test screen, change the **Delay** field to a smaller number, such as 12000 which is two seconds, and run a second 30 minute load test.
36. Start the performance test for gathering medium and high volatility statistics.
37. Monitor the life LDS and ensure that the LDS is no more than 50% of the maximum old generation. If the order rate causes the LDS to increase above 50%, increase the **Delay** field amount on the SoapUI Load Test screen.
38. Log in to the order management schema.
39. Enable incremental statistics on medium and high volatility orders.

```
execute om_db_stats_pkg.set_table_prefs_incremental(a_incremental =>
true,a_volatility => om_const_pkg.v_volatility_medium);
execute om_db_stats_pkg.set_table_prefs_incremental(a_incremental =>
true,a_volatility => om_const_pkg.v_volatility_high);
```

40. During the performance test, gather statistics on the medium and high volatility orders. The following statements also reduce the number of threads used for gathering statistics to two so that order processing does not suffer a performance impact.

```
execute DBMS_STATS.SET_SCHEMA_PREFS(user, 'DEGREE', 2);
execute om_db_stats_pkg.gather_order_stats(a_force => true,a_volatility =>
om_const_pkg.v_volatility_high);
execute DBMS_STATS.SET_SCHEMA_PREFS(user, 'DEGREE', 'DBMS_STATS.AUTO_DEGREE');

execute DBMS_STATS.SET_SCHEMA_PREFS(user, 'DEGREE', 2);
execute om_db_stats_pkg.gather_order_stats(a_force => true,a_volatility =>
om_const_pkg.v_volatility_medium);
execute DBMS_STATS.SET_SCHEMA_PREFS(user, 'DEGREE', 'DBMS_STATS.AUTO_DEGREE');
```

> **Note**
>
> If this is the first time you run this performance test with optimizer statistics gathered, some SQL execution plans may not be optimal. The database compares execution plans during the overnight maintenance window and accepts the better plans at that time. You may want to repeat the test the following day to see if there is a performance improvement. In addition, you may want your DBA to review which execution plans the database selected during the maintenance window because they may not always be the most optimal.

Determining the Sustainable Order Rate for a Managed Server

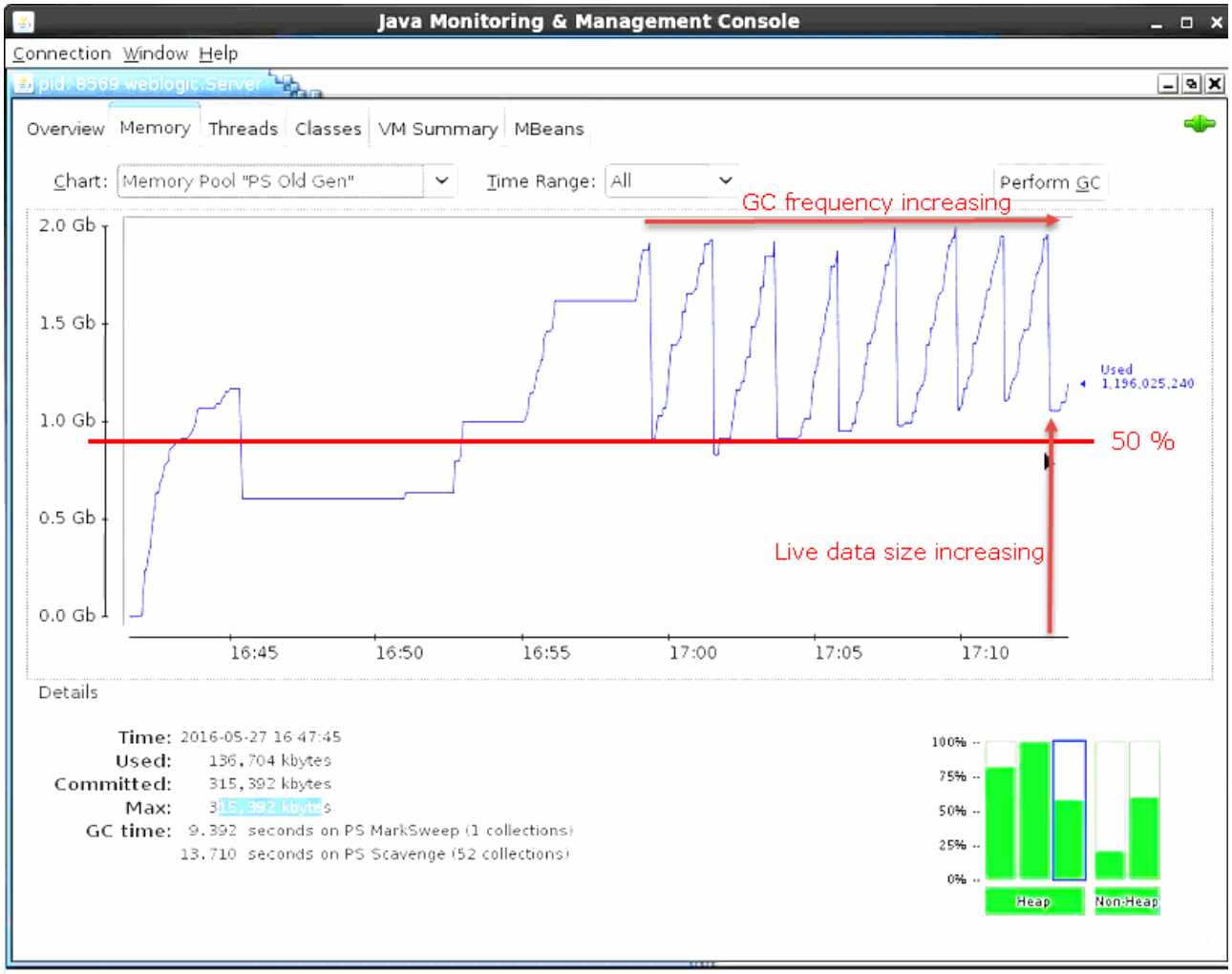
The following procedure should be done after completing warm up procedures.

1. Log in to the Oracle database as sys or the equivalent (traditional database or the pluggable database) and create an Oracle database automatic workload repository (AWR) snapshot.
2. Open the SoapUI load test.
3. In the **Limit** field enter a number. For example **3600**. This value is in seconds and causes the load test to run for 60 minutes, which is enough for the performance test although in some cases a longer period is required.
4. In the **Threads** field, enter **4**.
5. In the **Delay** field, enter a number. For example **15000**. This value represents a 15000 millisecond delay interval (2.5 seconds) between order submissions. The higher the value, the longer delay, and the less orders are submitted. The lower the value, the shorter the delay, and more orders are submitted.
6. Start the performance test.
7. As you are running the performance test, run JConsole.
8. Monitor the LDS and ensure that the level is stable at around 50% of the maximum old generation. If the order injection rate causes the LDS to increase above 50% for an extended time and if the LDS continues to increase so that garbage collection becomes more and more frequent, then decrease the number of orders you submit by increasing the **Delay** field amount on the SoapUI Load Test screen. For example, you might move from a 2.5 second delay to a 3 second delay.

   Figure 10-2 shows garbage collection with a steadily increasing frequency and LDS size. If the order injection rate were to remain at the current level, the managed server would eventually crash.

   The inverse scenario is also possible where the LDS size is lower than 50% and the frequency of garbage collection is much longer. In this case, you must increase the order injection rate by decreasing the Delay field amount on the SoapUI Load Test screen.

Figure 10-2 JConsole Garbage Collection Frequency and LDS Size



> **Note**
>
> The frequency of garbage collection and the size of the LDS for a 32 GB managed server is much larger than is depicted in Figure 10-2. The maximum old generation in a 32 GB managed server is 17 GB and the target LDS is 50% of the old generation which is 8.5 GB. Depending on the size and complexity of your orders, garbage collection make take a long time to occur. For example, garbage collection may only occur once every half hour. If very long garbage collection intervals are occurring, then increase the length of the performance test to two or even three hours to get an accurate garbage collection sampling. See step 3 to increase the length of the performance test.

9. When the LDS level has stabilized during the performance test, verify the number of automation threads being used to support the current number of orders that the managed server is processing. This value will be used to set the work manager maximum thread constraint for automations.

   To verify the number of automation threads in use and set the automation work manager maximum constraint, do the following:

a. Log in to the WebLogic Remote Console.

   The WebLogic Remote Console is displayed.

b. In **Edit Tree**, expand **Environment**. From **Environment**, select **Servers**.

   The Summary of Servers page is displayed.

c. Click the name of the WebLogic server that you are tuning.

   The configuration parameters for the server are displayed on a tabbed page.

d. Navigate to the **Monitoring Tree**. From the **Monitoring Tree**, select **Environment**. From **Environment**, select **Server**.

e. Click the **Threads** tab.

f. Navigate to **Monitoring Tree** From **Monitoring Tree**, select **Environment**.

g. Click **Scheduling**.

h. Select **Work Manager Runtime** field and enter **osmAutomationWorkManager**.

i. Refresh the screen every minute over 30 minutes to determine the highest number of active osmAutomationWorkManager threads you see after a refresh during that time.

> **Note**
>
> It is possible that there may be no active threads after refreshing. This does not indicate a problem unless the result occurs consistently.

10. Determine the maximum number of orders that can be in the JBoss and Coherence cache:

    a. Open a terminal on the machine running the manager server that you are tuning.

    b. In a text editor, open *domain_home***/servers/***managed_server***/logs/***managed_server***.out** (where *managed server* is the name of the managed server you are tuning).

    c. Search the log file for cache information using the **# Orders Information** text. Search through all instances and find the highest instance of the **Orchestration Cache** or the **Order Cache** value (whichever is greater of the two, although they are typically identical).

       This example, after searching through 30 cache information instances, the following cache information instance shows the highest set of cache values:

```
Cache                  % Full # Orders Information
---------------------- ------ -------- -----------
Closed Order Cache        12%   6/50
Historical Order Cache     8%   80/1000
Orchestration Cache       40%   400/1000
Order Cache               40%   401/1000
Redo Order Cache           0%   0/1000
```

       You would select the Order Cache number in this example because it is higher than the Orchestration Cache number.

11. After you finish the test, log in to the Oracle database again and create a second AWR snapshot.

12. Generate a report using both AWR snapshots.

13. Ask a database administrator (DBA) to analyze the AWR report to determine whether the database is performing as expected. DBAs are trained to detect database performance issues such as an undersized database or suboptimal SQL statement execution plans.

Tuning Work Manager Constraints and the Maximum Connection Pool Capacity

To set work manager constraints and the maximum connection pool capacity for OSM JDBC data sources, do the following:

> **Note**
>
> See the *OSM default Work Managers, Constraints related to upgrade process (Doc ID 3019290.1)* knowledge article on My Oracle Support, if you are upgrading from OSM version 7.3.1 or lower (source version) to OSM version 7.3.5 or higher (target version).

1. Log in to the WebLogic Remote Console.
2. In **Edit Tree**, expand **Environment**.
3. Navigate to **Scheduling**. From Scheduling, select **Max Threads Constraints**.
4. Select the **osmAutomationMaxThreadConstraint**.

   The Configuration tab appears.
5. In the **Count** field, enter the number of automation threads you observed in step 9 of the "Determining the Sustainable Order Rate for a Managed Server" procedure.
6. Click **Save**.
7. In Edit Tree, expand **Environment**. Navigate to **Scheduling**. From **Scheduling**, select **Max Threads Constraints**.

   The Summary of Work Managers page is displayed.
8. Select the **osmGuiMaxThreadConstraint**.

   The Configuration tab appears.
9. In the **Count** field, enter a number. This value is typically half of the osmAutomationMaxThreadConstraint count although if you have a solution that makes extensive use of manual tasks, you may need to raise this value.

   For example, if the osmAutomationMaxThreadConstraint was 60 then the osmGuiMaxThreadConstraint would be 30. However, if the OSM client users begin to experience long delays before they can gain access to an OSM client session, then you may want to raise this value. You should include OSM client users in you performance test to ensure that the ratio of threads allocated to osmGuiMaxThreadConstraint and osmAutomationMaxThreadConstraint is properly balanced.
10. Click **Save**.
11. In Edit Tree, expand **Environment**. Navigate to **Scheduling**. From **Scheduling**, select **Max Threads Constraints**.
12. Select the **osmJmsApiMaxThreadConstraint**.

    The Configuration tab appears.
13. In the **Count** field, enter a number. A typical starting value is between 4 to 6 which is enough in most cases. If the overall health of the managed server and the database is good, but the web service JMS queue is accumulating messages, then this value could be

increased. Increase the value incrementally, and run additional performance tests to ensure that the overall health of the managed server and database continues to be good.

To further protect the system from order rates that exceed the sustainable order rate for a managed server, it may be beneficial to trigger the OSM prioritization feature by reducing this value to cause JMS messages to queue up at the point where prioritization will have the greatest impact. In this context, a value of 1 can be considered if this value is still sufficient to maintain the sustainable order rate. If a value of 1 is still too high to trigger queuing when that rate is exceeded, consider configuring the equivalent of a fractional thread by associating an **osmJmsApiFairShareReqClass** fair share request class with the **osmWsJmsWorkManager** work manager, and setting the fair share value of the fair share request class to less than **50**.

14. Click **Save**.
15. In Edit Tree, expand **Environment**. Navigate to **Scheduling**. From **Scheduling**, select **Max Threads Constraints**.

    The Summary of Work Managers page is displayed.
16. Select the **osmHttpApiMaxThreadConstraint**.

    The Configuration tab appears.
17. In the **Count** field, enter a number. A typical starting value is between 4 to 6 which is enough in most cases. If the overall heath of the managed server and the database are good then this value could be increased. Increase the value incrementally, and run additional performance tests to ensure that the overall health of the managed server and database continues to be good.
18. Click **Save**.
19. Add all the maximum constraints you have configured and divide the total by 0.80 (80%) to determine the maximum connection pool size. For example, if you had the following constraint values:
    - osmAutomationMaxThreadConstraint = 60
    - osmGuiMaxThreadConstraint = 30
    - osmJmsApiMaxThreadConstraint = 5
    - osmHttpApiMaxThreadConstraint = 5

    which results in 100 / 0.80 which equals = 125 maximum connection pool size.
20. Log in to the WebLogic Remote Console.

    The WebLogic Remote Console is displayed.
21. In the **Edit Tree**, expand **Services**, then select **Data Sources**.

    The Summary of JDBC Data Sources screen is displayed.
22. Select an OSM JDBC data source. The OSM JDBC data source are as follows:

    osm_pool_*sid*_**group**_*y*

    where *sid* is the Oracle RAC database instance system identifier and *y* is the group letter.

    The Configuration tab is displayed
23. Click the **Connection Pools** tab.
24. In the **Maximum Capacity** field, enter the value you calculated in step 19.
25. Click **Save**.
26. Repeat steps 21 to 25 for all other OSM JDB connections.

27. Go to the shopping cart and click **Commit Changes**.

Tuning the JBoss and Coherence Maximum Order Cache

To set the maximum number of orders in the JBoss and Coherence cache for managed servers, do the following:

1. Using a text editor, open the *domain_home*/**oms-config.xml** file.
2. Change the JBoss OrderCacheMaxEntries value:

```
<oms-parameter>
    <oms-parameter-name>OrderCacheMaxEntries</oms-parameter-name>
    <oms-parameter-value>order_max</oms-parameter-value>
</oms-parameter>
```

   where *order_max* is the maximum number of orders that can be in the managed server's JBoss cache. Set this value to the number orders in the cache that you observed in step 10 of the "Determining the Sustainable Order Rate for a Managed Server" procedure.
3. Save and close the file.
4. Using a text editor, open the **osm-coherence-cache-config.xml** file.
5. Search on **osm-local-large-object-expiry**:

```
<scheme-name>osm-local-large-object-expiry</scheme-name>
     <eviction-policy>LRU</eviction-policy>
     <high-units>order_max</high-units>
```

   where *order_max* is the maximum number of orders that can be in the managed server's Coherence cache. Set this value to the number orders in the cache that you observed in step 10 of the "Determining the Sustainable Order Rate for a Managed Server" procedure.
6. Save and close the file.

Sizing the Redo Log Files

Using the AWR log files generated in the "Determining the Sustainable Order Rate for a Managed Server" section, check the log switches (derived) statistics. Log switching should occur at a frequency no less than 20 minutes apart. If the frequency of log switching is less than 20 minutes apart, then the redo log files are undersized. Increase the log file size, or number of redo groups, or both.

Checkpoint frequency is affected by several factors, including log file size and the **FAST_START_MTTR_TARGET** initialization parameter. If you set this parameter to limit the instance recovery time, Oracle Database automatically tries to checkpoint as frequently as necessary. The optimal size can be obtained by querying the **OPTIMAL_LOGFILE_SIZE** column from the **V$INSTANCE_RECOVERY** view. If **FAST_START_MTTR_TARGET** is not set, **OPTIMAL_LOGFILE_SIZE** is not set either.

For more information about sizing redo log files, see *Oracle Database Performance Tuning Guide*.

If the above change does not reduce checkpoint frequency, use the `renice` command to set the Log Writer Process (LGWR) to run at higher priority or run LGWR in the redo thread (RT) class by adding LGWR to the parameter: **_high_priority_processes='VKTM|LGWR"**. Only change **_high_priority_processes** in consultation with database support. For example, more processes may need to be added, such as PMON. And if the database is an Oracle RAC database, LMS should be added to this parameter. Test this change thoroughly.

Finally, if all other methods fail to reduce checkpoint frequency, set the **_log_parallelism_max** hidden parameter after consultation with database support.

Additional Performance Testing Options

The following sections provide additional performance testing options.

Performance-Related Features for Large Orders

In some cases, you might want to model large orders for OSM. A large order typically contains a sizeable payload with more than a hundred order items, and where each order item may contain many data elements. OSM provides the following features that can help you manage these large orders:

- Order automation concurrency control (OACC) is a policy driven OSM function that you can use to limit the number of concurrent automations plug-in instances that OSM can process at one time. For large orders, this ability can significantly reduce contention caused by an excessive number of automation plug-ins processing at the same time. High levels of automation plug-in contention can create performance issues because of the number of message retries and timeouts on the JMS queues. You can specify a policy using the **AutomationConcurrencyModels** parameter in the **oms-config.xml** file (see *OSM System Administrator's Guide*) or you can include an OACC policy in solution cartridge. See *OSM Developer's Guide* for information about creating OACC policies.
- Use the **oracle.communications.ordermanagement.table-layout.size** and the **oracle.communications.ordermanagement.table-layout.fetch-size** oms-config.xml parameter to create a threshold that limit the number of order rows that OSM can retrieve at one time from the database when using **Data** tab in the Order Management web client. See *OSM System Administrator's Guide* for more information.
- Use the **oracle.communications.ordermanagement.table-layout.threshold** oms-config.xml parameter to specify a threshold that automatically applies the style behavior table layout if a multi-instance node exceeds the threshold when using **Data** tab in the Order Management web client. See *OSM System Administrator's Guide* for more information.
- Ensure that the **show_all_data_history_logs_for_orderdetails** is set to false to reduce the number of logs that OSM generates. See *OSM System Administrator's Guide* for more information.

Distribution of High-Activity Orders

High-activity orders have a large number of processes, sub-processes, and tasks that must be run concurrently. Because the workload for a high-activity order can be significantly higher than for a typical order, OSM may redistribute a high-activity order to another active server instance proportionate to the managed server weights within the cluster. This redistribution based on weight ensures that one managed server does not get an unfair share of high-activity orders because of round robin-load balancing and ensures that high-activity orders are properly distributed among members in the cluster.

> **Note**
>
> A high-activity order is not exempt from order affinity: when OSM redistributes the order, it transfers the entire order and order ownership to another managed server. This redistribution does not mean that the order is being processed and owned by more than one managed server. See "About Order Affinity and Ownership in an OSM WebLogic Cluster" for more information.

High activity order processing is enabled by default. To tune or disable the high-activity order routing mode in OSM, you must configure a set of related parameters in the **oms-config.xml** file. See *OSM System Administrator's Guide* for a detailed reference of available parameters.

Measuring Order Throughput

Based on the order complexity guidelines specified in "Overview of Planning Your OSM Production Installation", you can calculate order throughput per second (TPS) using the following formula:

(throughput in task transitions per second) / (average number of tasks per order)

Throughput can then be calculated hourly, by multiplying by 3600 seconds per hour; or daily, by multiplying by 3600 seconds per hour plus the number of operating hours per day.

To determine a TPS value:

1. Log in to a database.
2. Enter the following statements:

```
alter session set nls_timestamp_format = 'dd-mon-yyyy hh24:mi:ss.ff3';

select
    count(*),
    min(timestamp_in),
    max(timestamp_in),
    (om_calendar_pkg.dsinterval_to_millis(max(timestamp_in)-min(timestamp_in))/1000)
duration,
to_char ((count(*)/(om_calendar_pkg.dsinterval_to_millis(max(timestamp_in)-
min(timestamp_in))/1000)),
'9999.999') tasks_per_sec
from om_hist$order_header
where hist_order_state_id = 4 and task_type in ('A','M','C') and
timestamp_in between 'dd-mon-yyyy hh24:mi:ss.ff3' and 'dd-mon-yyyyhh24:mi:ss.ff3';
```

   where:

   - *dd* is the day.
   - *mon* is the first three letters of the month.
   - *yyyy* is the year.
   - *hh24* is the number of hours in the 24 hour format.
   - *mi* is the number of minutes.
   - *ss* is the number of second.
   - *ff3* is the number of milliseconds.

Using the OM_ORDER_NODE_ANCESTRY Table

OM_ORDER_NODE_ANCESTRY is a table that stores the hierarchy of order group nodes. The table improves the efficiency and response time of worklist and order search queries, mainly for cartridges that have multi-instance subprocesses and a large number of flexible headers. For more information about parallel processes and multi-instance subprocesses, see the topic about understanding parallel process flows in *OSM Concepts*.

The downside of enabling the OM_ORDER_NODE_ANCESTRY table is increased CPU usage for order creation and updates, increased order creation response time, and most importantly increased disk usage. Specifically, OM_ORDER_NODE_ANCESTRY is one of the largest tables in OSM. It is often responsible for more than 20% of the space, depending on the depth of order templates, especially for large orders, such as O2A. Therefore, this table is disabled by default.

An Oracle database package called OM_ORDER_NODE_ANCESTRY_PKG contains the stored procedures that allow you enable and disable the OM_ORDER_NODE_ANCESTRY table.

> **Note**
>
> If you deploy cartridges with multi-instance subprocesses and are considering running OSM with the OM_ORDER_NODE_ANCESTRY table disabled, you must evaluate factors such as the ancestry depth in the master order template and the number of flexible headers, which could impact performance in the UI worklist and search results.

Table 10-1 shows the performance implications of running different cartridges in OSM with the OM_ORDER_NODE_ANCESTRY table enabled or disabled.

Table 10-1 Performance Implications of the OM_ORDER_NODE_ANCESTRY Table

OSM Solution	OM_ORDER_NODE_ANCESTRY Table Status	Performance Implications
Cartridges that do not require multi-instance subprocesses	Disabled	Positive impact: • Saves CPU time • Reduces order creation time • Improves throughput • Reduces OSM schema disk storage

Table 10-1 (Cont.) Performance Implications of the OM_ORDER_NODE_ANCESTRY Table

OSM Solution	OM_ORDER_NODE_ANCESTRY Table Status	Performance Implications
Cartridges that require multi-instance subprocesses	Enabled	Positive impact: • Improves response time when users retrieve worklist tasks and search orders Negative impact: • Increases CPU time • Increases order creation time • Degrades throughput • Increases OSM schema disk storage In this case, consider compressing the ancestry table. For more information about Oracle advanced compression, see Oracle Technology Network. Note that compression has the following negative impact: • Further increases order creation time • Increases SQL database CPU per execution (INSERTs)
Cartridges that require multi-instance subprocesses	Disabled	Positive impact: • Saves CPU time • Reduces order creation time • Improves throughput • Reduces OSM schema disk storage Negative impact: • Increases the response time when users retrieve worklist tasks and search orders In this case, Oracle recommends: • Avoiding deep order template node hierarchies • Eliminating unnecessary flexible headers

Enabling the OM_ORDER_NODE_ANCESTRY Table

When the OM_ORDER_NODE_ANCESTRY table is enabled, OSM populates the OM_ORDER_NODE_ANCESTRY table with data and uses queries on this table to support UI worklist and order searches. Running OSM in this mode is effective for new order id blocks. A new block is allocated when the current partition where new orders are created (known as the active partition) is exhausted.

Order ids are stored in the OM_ORDER_ID_BLOCK table. In this table, a column called ANCESTRY_POPULATED_UP_TO indicates the last order id in the block of order ids that has data in the OM_ORDER_NODE_ANCESTRY table.

An active order id block can be split logically, as in the following example:

- An order id block contains order ids from 0 to 100000. The order id block is NOT split yet and all order ids in this block contain ancestry data.
- An order id block is split. Orders ids between order id 0 and 2000 have ancestry data. Orders between order id 2001 and 100000 do not have ancestry data.

A block of order ids is **active** if it is the latest block for the current database instance (DBINSTANCE). The previous blocks for the database instance are **inactive** blocks.

When users retrieve worklist tasks or search for orders, OSM uses the data in order id blocks to determine if queries are run against the OM_ORDER_NODE_ANCESTRY table (old queries) or the OM_ORDER_INSTANCE table (new queries).

> **Note**
>
> Ancestry data is used only if the cartridge includes multi-instance tasks (pivot nodes).

You might need to switch several times between running OSM with the OM_ORDER_NODE_ANCESTRY table enabled and disabled. The following example scenarios illustrate circumstances that might necessitate switching between the two modes.

Scenario 1: Introducing multi-instance subprocess entities (enable, disable, enable)

1. You have upgraded OSM to a later version that includes this functionality. OSM continues to run with the OM_ORDER_NODE_ANCESTRY table enabled.
2. Because of large volumes of orders, you determine that OSM cartridges do not use multi-instance subprocesses and decide to disable the OM_ORDER_NODE_ANCESTRY table.
3. Some time later, you introduce multi-instance sub-process entities (for example, OSM needs to run a sub-process for each of the multiple addresses a customer has) by redeploying existing, or deploying new, cartridges.
4. You then determine that the worklist demonstrates performance degradation and decide to re-enable the OM_ORDER_NODE_ANCESTRY table.

Scenario 2: Eliminating multi-instance subprocess entities (disable, enable, disable)

1. You install the latest release of OSM, which includes this functionality. The OM_ORDER_NODE_ANCESTRY table is disabled.
2. You deploy a cartridge that uses multi-instance subprocesses, and leave the table disabled because performance test results are satisfactory.
3. You then determine that the worklist demonstrates performance degradation and decide to enable the OM_ORDER_NODE_ANCESTRY table.
4. Some time later, you redeploy updated cartridges so that all multi-instance subprocesses are eliminated. You then disable the OM_ORDER_NODE_ANCESTRY table.

You can run this procedure when OSM is online or offline.

To enable the OM_ORDER_NODE_ANCESTRY table:

1. Log in to SQL*Plus as the OSM core schema user.
2. Run the following command:

```
begin
        om_order_node_ancestry_pkg.enable_ancestry_table;
end;
```

Disabling the OM_ORDER_NODE_ANCESTRY Table

Running OSM with the OM_ORDER_NODE_ANCESTRY table disabled is suitable if you are deploying cartridges that do not include multi-instance subprocesses. When you run OSM with the table disabled, the OM_ORDER_NODE_ANCESTRY table is not populated and

hierarchical queries (for cartridges with multi-instance subprocesses) that are run using UI worklist or search functionality return ancestry data from the OM_ORDER_INSTANCE table.

You must disable the OM_ORDER_NODE_ANCESTRY table when OSM is offline because the procedure uses the last order id to split the block of order ids into two parts: populated and non-populated. For example, if the current order id is 100 and the last order id in the active block is 10000:

- [1...100...10000] is logically split into:
  - [1...100]: order ids with populated ancestry
  - [101...10000]: order ids with non-populated ancestry

To disable the OM_ORDER_NODE_ANCESTRY table:

1. Log in to SQL*Plus as the OSM core schema user.
2. Take the OSM server offline. For more information about stopping OSM, see *OSM System Administrator's Guide*.
3. Run the following command:

```
begin
       om_order_node_ancestry_pkg.disable_ancestry_table;
end;
```

> **Note**
>
> You can run the disable OM_ORDER_NODE_ANCESTRY table procedure only once on a single block of ids because the current block of ids can be split only once.

11

Upgrading to OSM 8.0

This chapter describes how to upgrade a traditional deployment of Oracle Communications Order and Service Management (OSM) to version 8.0. For information about moving from a traditional deployment of OSM to an OSM cloud native deployment and upgrading an OSM cloud native environment, see *OSM Cloud Native Deployment Guide*.

About OSM Upgrades

Upgrading OSM consists of the following process:

- Planning the upgrade
- Implementing and testing the upgrade on a development system
- Preparing to upgrade a production system
- Implementing and testing the upgrade on the production system

The upgrade process includes these tasks:

- Gather a list of components installed in your current OSM system
- Document the configuration selections made when installing your current OSM system
- Upgrade the platform (if applicable)
- Upgrade the Oracle WebLogic Server core software (if applicable)
- Upgrade or create a new WebLogic domain (if applicable)
- Update the WebLogic domain
- Upgrade the Oracle Database (if applicable)
- Upgrade the OSM software

The post upgrade process includes these tasks:

- Upgrade the development environment, including Oracle Communications Design Studio, Ant, OSM SDK, and OSM Tools
- Upgrade and redeploy cartridges to the OSM 8.0 server

For current version and patch information, see *OSM Compatibility Matrix*.

> **Note**
>
> If you are upgrading from OSM 7.4.1 to OSM 8.0, only upgrade the database if want to use the newer database version.

Supported Upgrade Paths

You can upgrade to OSM 8.0 from OSM 7.3.5.1.x release or later, and running on FMW 12.2.1.4.

> **Note**
>
> If you are upgrading OSM using a patch from My Oracle Support, you must read the patch Readme text and the contents of this chapter. In some cases the patch Readme may provide specific instructions that supersede those instructions included in this chapter.

About Backing Up Your Data

Before upgrading OSM, make a backup of your data files and the database. For general OSM backup and restore information, see *OSM System Administrator's Guide.* All of the backup information in that document is relevant to performing a backup prior to an upgrade.

OSM does not recreate or overwrite any existing WebLogic resources (including OSM users, groups and queues) except for the resources that were removed prior to upgrade (such as **oms.ear** and **cartridge_management_ws.ear**), which are upgraded if the existing version is lower than the version to which you are upgrading.

About Upgrading Oracle Database

The procedures in this section should be performed only by a qualified database administrator.

If you are performing an upgrade that requires a database version upgrade as well, you first upgrade the database and then upgrade OSM. In this case, the normal backup and restore procedures as discussed in OSM System Administrator's Guide still apply. It is recommended to back up the database both before and after upgrading it.

About OSM Customizations

Any customizations to views, tables, triggers, or other entities stored in the OSM database schema must be reapplied after OSM is upgraded.

Any custom reports on the OSM database schema must be reapplied or rewritten if schema changes are made to the newer database.

About Installer Disk Space

The amount of disk space required when upgrading is often higher than the 600 MB recommended for a new installation, and depends on the amount of disk space that is being used by the existing system. Oracle recommends that you test the upgrade in a non-production environment to determine the space required for your database.

To reduce the amount of disk space when upgrading, ensure that you purge orphan data and unnecessary orders, and drop unnecessary partitions, in order to use as little data as possible. For information about purging data and dropping partitions, see the topic about managing the OSM database schema in *OSM System Administrator's Guide*.

Preparing for an OSM Upgrade

The following sections describe how to prepare the OSM environment for an OSM upgrade.

Preparing the Environment

To prepare the OSM environment:

1. Ensure that all transactions are committed and all messages from relevant queues/topics are fully consumed or exported before the upgrade.
2. Stop all of the processes running against OSM (including the XML Import/Export application, XMLAPI agents and others). For more information about how to stop OSM, see *OSM System Administrator's Guide*.
3. Stop the WebLogic Server.
4. Back up the OSM database.

> ⚠ **Caution**
>
> The OSM installer stops the OSM schema upgrade if it encounters any issues. This can render the schema unusable. The OSM installer cannot rectify this problem on its own.

5. Do the following:
   - If you are creating a new WebLogic domain, back up the existing WebLogic server domain directory, custom SDK files, and scripts to a directory outside of your OSM installation.
   - If you are upgrading an existing OSM WebLogic domain, archive the *MW_home* directory including all subdirectories. Archive the *WLS_home* directory as well if it is not located in the *MW_home* directory.
6. Start the WebLogic server.
7. If you are upgrading from a version of OSM higher than 7.3.5.1.x, delete the OSM WebLogic Server application components, such as automation files (**plug-in.ear** file) from WebLogic using the Administration Console. The **oms.ear** file and **cartridge_management_ws.ear** files will get upgraded as part of the upgrade process.

   Also, manually delete these files from their respective directories located under *domain_home*/**servers**/*admin_server*/**upload**/ where *admin_server* is the name of the administration server for the domain.
8. If you are upgrading a development system that uses the Order-to-Activate cartridges and the Activation Integration Architecture emulators, remove the emulator **.ear** files from WebLogic using the console. Also, manually delete these files from under *domain_home***/servers/***admin_server***/upload/** where *admin_server* is the name of the administration server for the domain. You will have to re-deploy the emulator after the upgrade process is complete.
9. Stop the WebLogic Server.

Upgrading or Creating the WebLogic Domain

The following sections describe the steps to prepare the WebLogic domain for an OSM upgrade. You can:

- Upgrade the existing WebLogic domain. This is only supported if you are on OSM 7.3.5.1.x or later and running on FMW 12.2.1.4.

> **Note**
>
> If you're on a release earlier than OSM 7.3.5.1.x, or on 7.3.5.1.x or later and not running on FMW 12.2.1.4, you can either first upgrade to OSM 7.5 following the instructions in the OSM 7.5 *Installation Guide* and then follow this upgrade procedure, or create a new WebLogic domain.

- Create a new WebLogic domain.

Before you upgrade or create the WebLogic Server domain see the Fusion MiddleWare upgrade documentation and review the steps to:

- Upgrade custom security providers
- Upgrade Node Managers
- Upgrade WebLogic Domain
- Upgrade WebLogic Domain (remote managed servers)

Upgrading the WebLogic Domain of OSM 7.3.5.1.x, 7.4.0.0.3, or Higher to Fusion Middleware 14.1.2

The following section describes the steps to upgrade the WebLogic Domain of OSM 7.3.5.1.x, 7.4.0.0.3, or higher to Fusion Middleware 14.1.2.

To upgrade your WebLogic Domain, you need to first ensure that you complete the following prerequisite tasks:

1. Verify the Database User for the **WLSSchemaDataSource** Data Source. If your domain has the **WLSSchemaDataSource** data source, then you will need to verify which database user is assigned to it. If **<PREFIX>_WLS_RUNTIME** is assigned to it, then you need to change that to **<PREFIX>_WLS**.
2. Check **$DOMAIN_PATH** for a legacy **oms-config.xml** file. If it is present, then back it up and remove it before the upgrade so that the updated parameter values load correctly. You need to restart the Admin and Managed Servers immediately after the upgrade to apply the upgraded configuration.

The high-level steps for upgrading an OSM 7.3.5.1.x or later instance that is running on FMW 12.2.1.4 or higher, to Oracle Fusion Middleware 14.1.2 are the following:

> **Note**
>
> For more detailed instructions, follow the procedure that matches the upgrade for your domain in the Oracle Fusion Middleware document *Upgrading to the Oracle Fusion Middleware Infrastructure*.

1. Install the recommended JDK and required third-party software. For more information, refer to the *OSM Compatibility Matrix*.
2. Prepare the security store. The following steps use a basic OSM domain as an example:
   a. Ensure that you have performed the steps in Preparing the Environment, including stopping and deleting the **oms.ear** file. Deleting the **oms.ear** file prevents the previous version of the OSM application from running in the upgraded domain.

b. Stop all WebLogic servers and processes. For more information, refer to *Oracle Fusion Middleware Upgrading to the Oracle Fusion Middleware Infrastructure*.

3. Prepare to upgrade the security store by doing the following:

   a. In **domain_home/config/config.xml**, search for the **<jdbc-system-resource>** element that has a **<name>** child element where the value contains **pool** in the text. The **<jdbc-system-resource>** element will have another child element, **<descriptor-file-name>**. Make a note of both the name and the descriptor file name.

   b. In the **domain_home/config/jdbc/ directory**, look for the file that has a name that matches the value of the **<descriptor-file-name>** from the **config.xml** file. Note the value of the **<name>** element in that file.

   c. If the **<name>** elements in the two files differ, create a backup of the **domain_home/config/config.xml** file. Keep the original file safe, as you might need to restore it later during the process.

   d. Edit the **domain_home/config/config.xml** file so that the name of the **<jdbc-systemresource>** element **<name>** where the value contains the text **pool** matches the value of the **<name>** element in the file that matches **<descriptor-file-name>**.

4. Upgrade the security store by doing the following:

   a. Go to the **MW_home/oracle_common/upgrade/bin** directory for the new version of WebLogic and run Oracle Fusion Middleware **Upgrade Assistant** by using the following command for UNIX and Linux: `ua`.

   > **Note**
   >
   > Connect to the Upgrade Assistant as a user without sysdba privileges.

   The **Welcome** screen of the **Upgrade Assistant** is displayed.

   b. Click **All Schemas** and then select **All Schemas Used By a Domain**.

   c. In the **Available Components** page, ensure that you select **Common Infrastructure Services**, **Oracle Audit Services**, and **Oracle Platform Security Services** in order to upgrade the OPSS and IAU schemas. For more information, refer to Oracle Fusion Middleware *Upgrading to the Oracle Fusion Middleware Infrastructure*.

5. If you edited the **domain_home/config/config.xml** file in step 3, replace the version that you edited with the original version that you saved.

6. After the schemas are upgraded, reconfigure the domain by doing the following:

   a. Go to the **MW_home/oracle_common/common/bin** directory for the new version of WebLogic and run the Reconfiguration Wizard by using the following command: `reconfig.sh/cmd`.

   b. In the **Advanced Configuration** screen, select **Deployments and Services** and choose to create missing schemas - WLS schema. The missing WLS schema can be updated using the **Upgrade Assistant**. For more information, refer to Oracle Fusion Middleware *Upgrading to the Oracle Fusion Middleware Infrastructure*.

> **Note**
>
> In the **Deployments Targeting** screen, ensure that all the required libraries and applications are targeted to the OSM cluster and that changes are not required for other servers. In the **Service Targeting** screen, besides the OSM data-source to OSM cluster/server, ensure that all non-OSM data sources that are used by the domain are deployed into the Administration server, OSM cluster/server and proxy server. For proxy server configuration, the built-in HTTP Proxy is deprecated. Oracle recommends using external load-balancing solutions instead, such as Oracle HTTP Server, Apache HTTP Server, and hardware load balancers.

7. Upgrade component configurations by doing the following:
   a. Go to the **MW_home/oracle_common/upgrade/bin** directory for the new version of WebLogic and run Oracle Fusion Middleware **Upgrade Assistant** by using the following command ua.
   b. Select **All Configurations Used by a Domain**.
   c. After the **Upgrade Assistant** successfully completes the WebLogic component configurations operation, start the servers and verify in WebLogic Remote Console that all the relevant servers, applications, and libraries are upgraded to the latest version. For information about packing and unpacking the upgraded domain to distribute to other machines in a clustered environment, refer to Replicating the Domain Template on Other Machines.

> **Note**
>
> After the domain upgrade finishes, and before you install OSM 8.0, you must use WebLogic Remote Console to go to the JMS Connection Factories (for example, **oms_connection_factory**) to ensure that the **Default Targeting Enabled** check box is not selected. If it is selected, deselect it and then restart the servers (if they are running).

Creating a New WebLogic Domain

To create a new WebLogic domain with ADF:

> **Note**
>
> If your environment is earlier than OSM 7.3.5.1.x, or 7.3.5.1.x or later and not running on FMW 12.2.1.4, you must create a new WebLogic domain with ADF. Review the existing hardware and ensure it is supported by OSM 8.0. Otherwise, you must upgrade to supported hardware. For more information, see "General Hardware Sizing and Configuration Recommendations."

1. Install the recommended JDK version. See "*OSM Compatibility Matrix*" for details on the recommended version.
2. Install the Oracle Fusion Middleware Infrastructure version for OSM 8.0 (including Oracle WebLogic Server and ADF).
3. Apply any required Oracle WebLogic Server and ADF patches.

4. Create a new domain. Use the same domain name as the existing domain. When creating the new domain, Oracle recommends that you select **Oracle Enterprise Manager** template, in order to view and manage OSM logs. Also, you must select **Oracle JRF**. This is required for OSM as it makes use of ADF. See "Installing and Configuring the WebLogic Server Cluster."

> **Note**
>
> When you select the **Oracle JRF** template, the template for **WebLogic Coherence Cluster Extension** is also selected. Do not deselect the coherence cluster extension template option.

Re-create user accounts and settings. You must recreate all the users and other settings that you had configured in your old domain in your new domain.

> **Note**
>
> It is important to create user accounts and settings *before* performing the OSM upgrade, to prevent accidental deletion of user information from the OSM database. When OSM starts, it checks the users in the database schema against the users configured in WebLogic Server and deletes invalid users from the database. Oracle recommends that, prior to the OSM upgrade, you export all of the security realm data from your existing domain and import that data into your new domain. See *Oracle Fusion Middleware Administering Security for Oracle WebLogic Server* for information about migrating security data.

Note that password requirements may have changed with the newer version of Oracle WebLogic. If you are upgrading a system that uses the Order-to-Activate cartridges, users and settings to support the cartridges will have to be recreated. See "Updating Order-to-Activate Cartridges."

If your previous domain was in a clustered WebLogic Server environment re-create the same environment. See "Installing and Configuring the WebLogic Server Cluster" for information about installing OSM in a clustered environment.

Updating the WebLogic Domain

The following section describes WebLogic domain updates that may be required.

Updating JMS Security Policy Settings

Starting with OSM 7.0.1, JMS Queues or Topics that are part of the JMS system module (**oms_jms_module**) have a security policy. No action is required for applications built using the OSM SDK, but modifications are required to external (non-OSM) applications that communicate with OSM using JMS.

An application that communicates with OSM using JMS already has **ejb-jar.xml** and **weblogic-ejb.jar.xml** files in the code. Before these applications are used with OSM 7.0.1 or later, the following additions must be made.

In the **ejb-jar.xml** file, add a <security-identity> for each EJB. In the example below the EJB is a message-driven bean.

```
<message-driven>
   ...
   <security-identity>
      <run-as>
         <role-name>test-role</role-name>
      </run-as>
   </security-identity>
</message-driven>
```

In the **ejb-jar.xml** file, include the <security-role> as a part of the <assembly-descriptor>:

```
<assembly-descriptor>
   <security-role>
      <description/>
         <role-name>test-role</role-name>
      </security-role>
   ...
</assembly-descriptor>
```

In the **weblogic-ejb.jar.xml** file map the <security-role> to the oms-automation user to allow EJBs to access the JMS resources:

```
<weblogic-ejb-jar>
   ...
   <security-role-assignment>
      <role-name>test-role</role-name>
      <principal-name>oms-automation</principal-name>
   </security-role-assignment>
   <run-as-role-assignment>
      <role-name>test-role</role-name>
      <run-as-principal-name>oms-automation</run-as-principal-name>
   </run-as-role-assignment>
</weblogic-ejb-jar>
```

Upgrading the Database

If you are currently running on a 12*c* (12.1.0.2) database, export the OSM data, upgrade the database to a supported version for OSM 8.0 and import the OSM data. See "Software Requirements" for version details and information on required patches.

Refer to the Oracle Database documentation for information about upgrading Oracle Database Server and migrating data.

Updating Coherence Properties for Managed Servers

The coherence cluster property for non-OSM managed servers might all be set to the same default value, which creates a conflict in the environment.

If you are upgrading OSM in a non-clustered environment, you must update coherence properties from the default, after you upgrade the domain and before you run the installer.

To update coherence properties in a non-clustered environment:

1. Log in to the WebLogic Remote Console.

   The Remote Console is displayed.

2. In the **Edit Tree**, expand **Environment**. From **Environment**, click **Servers**.

   The Summary of Servers window is displayed.

3. Select **Server**. From **Server**, select **Advanced Tab**. From **Advanced Tab**, select **Coherence** sub tab.
4. Do one or both of the following:
   - On the Coherence tab for each non-OSM managed server, set the **Coherence Cluster** property to **None**.
   - On the Coherence tab for the OSM managed server, set the **Coherence Cluster Unicast Listen Port** property to a unique port.
5. Click **Save**.
6. Click the shopping cart and select **Commit Changes**.

Upgrading the SDK Library Names

If you are upgrading to OSM 8.0 from OSM 7.5 or prior, the installation includes changes to the names of some of the SDK library JAR files. If you have a custom implementation that has SDK library references in the classpath, you must change these references so that they use the new JAR file names.

Table 11-1 lists the old file names that you must replace with the new file names in classpath references.

Table 11-1 SDK Third-Party Library JAR File Names

Old File Name	New File Name
commons-codec-1.9.jar	commons-codec.jar
commons-collections4-4.0.jar	commons-collections4.jar
commons-io-2.4.jar	commons-io.jar
commons-lang-3.3.1.jar or commons-lang3-3.3.2.jar	commons-lang3.jar
commons-logging-1.1.1.jar	commons-logging.jar
commons-net-3.3.jar	commons-net.jar
commons-pool2-2.2.jar	commons-pool2.jar
commons-vfs2-2.0.jar	commons-vfs2.jar
httpclient-4.1.2.jar or httpclient-4.3.5.jar	httpclient5.jar
httpcore-4.3.2.jar	httpcore5.jar httpcore5-h2.jar sl4j-api.jar
jaxen-1.1.3.jar or jaxen-1.1.6.jar	jaxen.jar
ojdbc6.jar or orojdbc7.jar	ojdbc11.jar
resolver.jar	xml-resolver.jar xmlresolver.jar
saxon9-dom.jar or saxon-license.lic	saxon-ee-java-osm-license.jar

Table 11-1 (Cont.) SDK Third-Party Library JAR File Names

Old File Name	New File Name
saxon9-xpath.jar saxon9.jar saxon9ee.jar	saxon-ee-java.jar
text-table-formatter-1.1.1.jar	text-table-formatter.jar
xschema.jar	(no longer needed as a separate library, now part of xmlparserv2.jar)
tools.jar	(no longer needed as a separate jar, as from JDK 9 onwards its compiler and tool classes are built into the standard JDK modules)

Upgrading OSM to 8.0

This procedure describes how to upgrade OSM to version 8.0.

> ⚠ **Caution**
>
> Before upgrading, refer to "About OSM Upgrades" for important preparation steps.

Performing the OSM Application Upgrade

OSM supports the following upgrade scenarios:

- Upgrade from previous release of a WLS cluster.
- Upgrade from previous release of standalone WLS.

The upgrade will be performed in online mode for which your AdminServer should be up and running.

Before upgrading the OSM application, ensure that you have completed all prerequisite procedures described in the sections preceding this section. These prerequisite tasks include upgrading the database and Fusion Middleware.

To upgrade the OSM application, complete the following steps

1. Capture new configuration properties by running the discovery script.
2. Upgrade OSM Database Schema.
3. Upgrade OSM Configuration in WebLogic domain.
4. Post-upgrade Activities.

Capturing New Configuration Properties
To capture the new configuration properties:

1. Start the WebLogic server in the new or upgraded WebLogic domain.
2. Take a backup or, preferably manage your **OSM_CFG_HOME** directory using source control mechanisms.
3. Log into the OSM DB using SQL*Plus as sysdba and run the following:

```
grant create any context to sysuser as sysdba with admin option
```

4. Run the following interactive script to capture the updated properties:

```
$ export PASSPHRASE=passphrase
$ export OSM_INSTALLER_HOME=path_to_installerhome
$ OSM_INSTALLER_HOME/scripts/discover.sh -n osm_env_name -c $OSM_CFG_HOME
```

For detailed description about the properties, refer to "Specifying Configuration Properties in the Configuration Phase".

Upgrading OSM Database Schema and OSM Configuration Together
You can upgrade the OSM Database schema and OSM configuration in WebLogic domain together by running a single command. This will first upgrade the OSM DB schema and then the OSM Configuration in weblogic domain.

```
$ export PASSPHRASE=passphrase #It should be same as used in discover.sh
$ export OSM_INSTALLER_HOME=/path/to/installerHome
$ export FMW_HOME=/path/to/fmw_home

$ $OSM_INSTALLER_HOME/scripts/configOSM.sh -n osm_env_name -c $OSM_CFG_HOME
```

Alternatively, you can upgrade them one by one. Refer to the **Installing DB Schema and OSM Separately** section that follows.

Installing DB Schema and OSM Separately
You have the option to upgrade OSM DB schema and OSM Configuration separately. This is not required if you have already performed the step for upgrading OSM DB schema and OSM Configuration together described above.

Upgrade OSM Database Schema
Regardless of a new OSM schema installation or an upgrade of an existing OSM schema, invoke the **configDB.sh** script:

```
$ export PASSPHRASE=passphrase #It should be same as used in discover.sh
$ export OSM_INSTALLER_HOME=/path/to/installerHome

$ $OSM_INSTALLER_HOME/scripts/configDB.sh -n osm_env_name -c $OSM_CFG_HOME
```

The installer performs the required database operations automatically based on the **configuration.properties** captured in the above step.

The OSM installer script identifies if the database already contains an existing OSM schema. If the database contains OSM schema, the script upgrades the schema to the version of the installer. This can be invoked even if the schema is already upgraded - it recognizes the validity of the schema and affects no changes.

Upgrade OSM Configuration in WebLogic Domain
Run the following script to upgrade the **osm.ear** and **cartridgemanagement.ear** files to the existing domain in online mode. Here, only the AdminServer should be up and running.

```
$ export PASSPHRASE=passphrase #It should be same as used in discover.sh
$ export OSM_INSTALLER_HOME=/path/to/installerHome
$ export FMW_HOME=/path/to/fmw_home

$ $OSM_INSTALLER_HOME/scripts/configDomain.sh -n osm_env_name -c $OSM_CFG_HOME
```

> **Note**
>
> You will have to pass the **$OSM_INSTALLER_HOME** and **$FMW_HOME** as command line arguments (-l and -f respectively) if these environment variables are not set before invoking the above scripts. It is recommended to always store the **PASSPHRASE** in an environment variable. You can use the default variable name of **PASSPHRASE** and invoke the above scripts without the -p parameter. Alternatively, you can choose an environment variable name to store the **PASSPHRASE** in, and pass that variable name as a command line argument (-p).You must ensure all the installer scripts that need to be invoked for this environment are given the same passphrase value.

When managed servers are installed on hosts separate from the admin server, their server tmp, cache and stage directories should be cleaned up after the upgrade is done. Those directories can be found under **$DOMAIN_HOME/servers/*MS_Name***
where **$DOMAIN_HOME** is the Weblogic domain location. After cleaning up the directories,restart the admin server and then start up all the other managed servers.
Post-Upgrade Activities

1. In a new window, log into SQL*Plus as the OSM database user and run the procedure below in the primary database schema to gather schema statistics:

   ```
   BEGIN
   DBMS_STATS.GATHER_SCHEMA_STATS (
   OWNNAME=>USER,
   ESTIMATE_PERCENT=>10,
   GRANULARITY =>'ALL',
   CASCADE =>TRUE,
   BLOCK_SAMPLE=>TRUE);
   END;
   ```

   Because the upgrade to OSM 8.0 involves data movement, it is important that database statistics be updated. If you plan to verify the installation prior to daily automatic statistics collection, use the DBMS_STATS.GATHER_SCHEMA_STATS procedure to gather statistics manually.

   Oracle recommends setting the ESTIMATE_PERCENT parameter of GATHER_SCHEMA_STATS to DBMS_STATS.AUTO_SAMPLE_SIZE to maximize performance gains while achieving necessary statistical accuracy. With a larger ESTIMATE_PERCFENT value, statistics gathering for a large database could take several hours. If you prefer a faster, less accurate result, use a small ESTIMATE_PERCENT value such as 1.

   You can continue with the upgrade procedure while this statistics gathering runs.

2. Reapply any **oms-config.xml** file customizations from the previous release. This file is located in the server startup directory, which is usually the home directory for the domain.
3. Reapply any other EAR file customizations from the previous release by undeploying the newly installed **oms.ear** file and redeploying an **oms.ear** file containing the required customizations.
4. Reapply customizations to views, tables, triggers, or other entities stored in the OSM database schema.
5. Reapply or rewrite any custom reports if schema changes were made to the newer database.

6. If a new WebLogic domain was created, copy the contents of the **Attachments** directory from the old WebLogic domain to the newly created WebLogic domain. Ensure that the file permissions allow the OSM application to read and write the files copied to the new domain.
7. Shutdown and restart the WebLogic server in the new or upgraded domain.
8. Refer to *Design Studio Installation Guide* for any required upgrade actions for Design Studio.
9. Upgrade your cartridges. Refer to Upgrading Pre-7.3.5 Cartridges to OSM 8.0 for more information.

When the upgrade is complete, instruct your OSM web client users to clear their browser's temporary cache files. Refer to the web browser's documentation for information about how to do this.

> **Note**
>
> If you do not clear your cache, you may experience unexpected errors while using the upgraded OSM web client because the web browser may still be using the previous versions of certain OSM web client files.

Recovering from a Database Upgrade Failure

If an error occurs while the installer is upgrading the database schema, you can fix the error and run the installer again. The installer then resumes the upgrade from the point of failure, which means you do not have to roll back the entire upgrade and start from the beginning.

For the procedure for upgrading OSM, see "Performing the OSM Application Upgrade ."

When you run the installer, the installer generates and then stores an upgrade plan. When you re-run the installer after a failure, this plan is run one action at a time. In general, each migration script and SQL statement that modifies the schema is a separate action and database transaction.

The following are the high-level steps in the process of recovering from a database upgrade failure. For information about handling an OSM database schema installation failure, see"Troubleshooting OSM Installation Problems - Handling an OSM Database Schema Installation Failure."

1. Finding the Issue that Caused the Failure
2. Fixing the Issue that Caused the Failure
3. Restarting the Upgrade from the Point of Failure

Finding the Issue that Caused the Failure

See "Handling an OSM Database Schema Installation Failure" for details.

Fixing the Issue that Caused the Failure

Use the information in the log or error messages to fix the issue before you restart the upgrade process. For information about troubleshooting log or error messages, see *OSM System Administrator's Guide*.

Restarting the Upgrade from the Point of Failure

In most cases, restarting the upgrade consists of pointing the installer to the schema that was partially upgraded, and then rerunning the installer.

Keep the following in mind when preparing to restart an upgrade:

- Most migration actions are a single transaction, which is rolled back in the event of failure. However, some migration actions involve multiple transactions. In this case, it is possible that some changes were committed.
- Most migration actions are repeatable, which means that they can safely be re-run even if they were committed. However, if a failed action is not repeatable and it committed some changes, either reverse all the changes that were committed and set the status to **FAILED**, or complete the remaining changes and set the status to **COMPLETE**.

To restart the upgrade after a failure:

1. Determine which action failed and why by using the information in "Fixing the Issue that Caused the Failure."
2. If the status for the failed action is **STARTED**, check the database to see whether the action is finished or still in progress.
3. If the failed action is still in progress, either end the session or wait for the action to finish (roll back).

   > **Note**
   >
   > The transaction might not finish immediately after the connection is lost, depending on how fast the database detects that the connection is lost and how long it takes to roll back.

4. Fix the issue that caused the failure.

   > **Note**
   >
   > If the failure is caused by a software issue, contact Oracle Support. With the help of Oracle Support, determine whether the failed action modified the schema and whether you must undo any of those changes.
   >
   > If you decide to undo any changes, leave the action status set to **FAILED** or set it to **NOT STARTED**. When you retry the upgrade, the installer starts from this action. If you manually complete the action, set its status to **COMPLETE**, so that the installer starts with the next action. Do not leave the status set to **STARTED** because the next attempt to upgrade will not be successful.

5. Restart the upgrade by running the installer.

   The installer restarts the upgrade from the point of failure.

XQuery Model Changes

The latest version of Saxon has a stricter approach to type checking. There are changes to how Saxon converts arguments and returns types to and from Java functions, so some XQuery functions that were written for earlier versions of Saxon might not work correctly.

Using java.util.Map as an Argument or Return Type

XPath 2.0 introduced Map as an internal data type. If a variable of type `java.util.Map` is not explicitly typed, Saxon casts the variable to the native XPath Map type. The solution is to explicitly bind any `java.util.Map` variables to the Java namespace. For example,

Before:

```
declare namespace osm="http://xmlns.oracle.com/communications/ordermanagement/model";
declare namespace ns1="http://www.metasolv.com/OMS/OrderModel/2002/06/25";

declare namespace mapUtil = "java:java.util.Map";

declare variable $mvmap external;

declare variable $mvkey as xs:string* external;

<model:modelVariable name="{/ns1:model/ns1:cartridge/@namespace}"
namespace="SystemDefinedNamespace" xmlns:model="http://xmlns.oracle.com/communications/
ordermanagement/model">
  <model:description>model variable</model:description>
  {
    for $i in ($mvkey)
     let $r := mapUtil:get($mvmap,$i)
    return
      <model:entry name="{$i}"  >
        <model:value>{$r}</model:value>
      </model:entry>
   }
</model:modelVariable>
```

After:

```
declare namespace osm="http://xmlns.oracle.com/communications/ordermanagement/model";
declare namespace ns1="http://www.metasolv.com/OMS/OrderModel/2002/06/25";

declare namespace jt="http://saxon.sf.net/java-type";
declare namespace mapUtil = "java:java.util.Map";
declare variable $mvmap as jt:java.util.Map external;

declare variable $mvkey as xs:string* external;

<model:modelVariable name="{/ns1:model/ns1:cartridge/@namespace}"
namespace="SystemDefinedNamespace" xmlns:model="http://xmlns.oracle.com/communications/
ordermanagement/model">
  <model:description>model variable</model:description>
  {
    for $i in ($mvkey)
     let $r := mapUtil:get($mvmap,$i)
    return
      <model:entry name="{$i}"  >
        <model:value>{$r}</model:value>
      </model:entry>
```

```
   }
</model:modelVariable>
```

Using java.util.Collection as a Return Type

If a Java function returns a collection or array, Saxon converts this to an XPath sequence. The return object can no longer be passed to a `java.util.Iterator`. The latest version of Saxon makes it easier to deal with collections. Instead of using a Java iterator, you can use the XPath sequence. For example,

Before:

```
declare function local:getAttachedProvOrderEBM() as element()*
{
   let $names := context:getAllAttachmentFileNames($context)
   return
      if (fn:exists($names)) then
      (
          let$name := iterator:next(collection:iterator($names))
          return
              if (fn:exists($name)) then
              (
                  saxon:parse(saxon:base64Binary-to-string(saxon:octets-to-
              )
              else ()
      )
      else ()
};
```

After:

```
declare function local:getAttachedProvOrderEBM() as element()*
{
   let $names := context:getAllAttachmentFileNames($context)
   return
      if (fn:exists($names)) then
      (
          let$name := [1]($names))
          return
              if (fn:exists($name)) then
              (
                  saxon:parse(saxon:base64Binary-to-string(saxon:octets-to-
              )
              else ()
      )
      else ()
};
```

Invoking Overloaded Methods of Same Number of Arguments with Ambiguous Types

There are extensive changes to how overloaded methods are chosen. In most cases, these changes are transparent. But in the case of overloaded methods with the same number of arguments, if the supplied argument has an ambiguous type, Saxon is unable to resolve the method to invoke, for example, Invoking java.lang.String.valueOf() with a byte[]. The solution in these cases is to cast the argument to a proper Java type. For example,

Before:

```
saxon:parse(xs:string(javaString:valueOf(context:getAttachment($context,
xs:string($name)))))/provord:ProcessProvisioningOrderEBM
```

After:

```
saxon:parse(xs:string(javaString:valueOf(context:getAttachmentAsString($context,
xs:string($name)))))/provord:ProcessProvisioningOrderEBM
```

Handling Null Values from Java Functions

If a Java method returns null, the XPath value is an empty sequence. If the XQuery script does not correctly handle empty sequences, an XPath exception is thrown. You must ensure in the XQuery that invocations of Java functions that can return null are properly handled to check for empty sequences.

Additional Configuration for JMS Service Migration

If you are upgrading OSM instances in a clustered environment that make use of JMS service migration, you must perform the following procedure.

> **Note**
>
> Oracle recommends that you use whole server migration, rather than JMS service migration, to address your high-availability needs. For more information, see "Understanding Whole Server Migration for High Availability."

To perform additional configuration for JMS service migration:

1. Log in to the WebLogic Remote Console.

   The Remote Console is displayed.
2. From the **Edit Tree**, click **Services**. From **Services**, select **JMS Modules**.

   The Summary of JMS Modules screen is displayed.
3. Click **oms_jms_modules**. From **oms_jms_modules**, select **Uniform distributed Queues.**

   List of available resources are displayed

   The Settings for oms_jms_module screen is displayed.
4. For each instance of the **oms_cartridge_deploy**_*managed_server* queue (where *managed_server* is one of the managed servers in the cluster) do the following:
   a. Click **oms_cartridge_deploy**_*managed_server*.

      The Settings for **oms_cartridge_deploy**_*managed_server* screen is displayed.
   b. Click the **Configuration** tab then the **General** subtab.
   c. From the **Templates** list, select **None**.
   d. Click **Save**.
   e. Click the **Subdeployment** tab.
   f. From the Subdeployment list, select the **omsJmsNonMigratableServer**_*managed_server* that corresponds to the same managed server that appears in the **oms_cartridge_deploy**_*managed_server* queue name.

For example, oms_cartridge_deploy_OSMServer1 should use the osmJmsNonMigratableServer_OSMServer1 subdeployment.

g. Click **Save**.

h. Click the **Configuration** tab then the **General** subtab.

i. From the Template list, select **omsJmsNonMigratableTemplate**_*managed_server*. that corresponds to the same managed server that appears in the **oms_cartridge_deploy**_*managed_server* queue name.

For example, oms_cartridge_deploy_OSMServer1 should use the omsJmsNonMigratableTemplate_OSMServer1 template.

j. Click **Save**.

k. Restart the managed server that the **oms_cartridge_deploy**_*managed_server* queue is associated with.

5. After you restart each managed server, log in to the OSM Order Management web client.
6. Click **Refresh Cache**.

The **Confirm Metadata Refresh** dialog box appears.

7. Click **Yes**.

Upgrading the Development and Administration Environment

OSM supports a development and administration environment that includes the following software components that may require an upgrade:

- **Design Studio**: The currently supported versions of the Design Studio core and plug-ins may be different than those certified with your source OSM release. There may also be additional plug-ins that were not available with your OSM source release. See *Design Studio Compatibility Matrix* for Design Studio for Order and Service Management compatibility information.
- **Ant**: Apache Ant is a Java tool that is required by the OSM XML Import Export application (XMLIE) and the OSM cartridge management tool (CMT) used for deploying custom task assignment algorithms. You may need to upgrade the Ant version to use these applications. See *OSM Compatibility Matrix* for version details.
- **OSM SDK Tools and Samples**: You can install new versions of the OSM SDK tools and samples using the OSM installer. See "Upgrading OSM to 8.0" for more information about running the OSM installer.

Upgrading Pre-7.3.5 Cartridges to OSM 8.0

After you have upgraded your OSM system to version 8.0, perform the following procedure to enable your pre-OSM 7.3.5 cartridges to run in the newly upgraded environment.

> **Note**
>
> Upgrade your cartridges only if they were built using an OSM SDK older than version 7.3.5. If you need to build or re-build cartridges for OSM 8.0, use the OSM 8.0 SDK and Design Studio target OSM version as 8.0.0.0.0 Use the Design Studio target OSM version that is closest to the OSM version.

> **Warning**
>
> It is mandatory to upgrade or rebuild cartridges whose Java plugins or emulators currently depend on JDK 8, or earlier and Java EE 7, or earlier libraries. This ensures that all such components are migrated to use Java 21 and Java EE 8 libraries for compatibility with FMW 14.1.2 and Java 21.

Cartridge Upgrade Prerequisites

Before you upgrade cartridges, ensure the following prerequisites are met:

- OSM is upgraded to 8.0.
- Design Studio is upgraded to the required version. See "Upgrading the Development and Administration Environment."
- The OSM 8.0 SDK component is installed.
- The Oracle WebLogic Server domain for OSM 8.0 is running.
- Old existing domain configurations, such as JMS queues, users, groups, and emulators have been recreated in the new domain.
- Automation plug-in **.ear** files are deleted from WebLogic using the console.
- All cartridges are backed up.

Cartridge Upgrade Procedure

To upgrade your pre-7.3.5 cartridges to OSM 8.0:

1. Set global default Java preferences.

   In Design Studio, from the **Window** menu, select **Preferences**. Then expand **Java** and do the following:

   a. Select **Installed JREs** and check the box next to the Java 21 JRE you are using.

   b. Expand **Installed JREs**, select **Execution Environments**, select **JavaSE-21** from the **Execution Environments** list, and select the newly installed JRE from the **Compatible JREs** list.

   c. Select **Compiler** and set the **Compiler Compliance Level** to **21**.

2. Set global default OSM preferences.

   In Design Studio, from the **Window** menu, select **Preferences**. Then expand **Oracle Design Studio** and select **Order and Service Management Preferences**. Set the following values:

   - OSM SDK Home: Specify the location of the unzipped OSM 8.0 SDK.

> **Note**
>
> If you have customized your build path, ensure that your environment is looking for **automation_plugins.jar** in the following directory:
>
> `$OSM_SDK/Automation/automationdeploy_bin`
>
> Here, `$OSM_SDK` is the path of OSM SDK, which you would have downloaded and unzipped.

For information about automation plug-in dispatch modes and how performance is improved, see *OSM Developer's Guide*.

3. Set cartridges to use the correct build path.

   For each cartridge, do the following:

   a. In the Studio Projects view, right-click the cartridge and select **Properties**.

   b. Select **Java Build Path** and click the **Libraries** tab.

   c. Select the **JRE System Library** from the list and click **Edit**.

   d. Select **Execution Environment** and select the option that begins **JavaSE-21** and references the location where you installed Java 21.

   e. Click **OK**.

   f. If **automation_plugins.jar** is present in the list on the **Libraries** tab, select it and click **Migrate JAR File**. Select the **automation_plugins.jar** file in the following directory:

      `$OSM_SDK/Automation/automationdeploy_bin`

   g. Click **Finish**.

   h. Click **OK** in the Properties window.

4. Set cartridge management variables.

   For individual cartridges in your workspace, open the Project editor **Cartridge Management Variables** tab and set the following variables according to your requirements:

   - Set FAST_CARTRIDGE_UNDEPLOY to **true** to undeploy a cartridge from OSM without purging cartridge metadata or order data, or set it to **false** to purge cartridge metadata and order data during the undeploy operation.
   - Set PURGE_CARTRIDGE_BEFORE_DEPLOY to **true** to undeploy the previous version of a cartridge before deploying the new version, or set it to **false** to update the cartridge with the changes for the new version. If PURGE_CARTRIDGE_BEFORE_DEPLOY and FAST_CARTRIDGE_UNDEPLOY are set to **true**, the cartridge is undeployed using the fast undeploy functionality before it is redeployed. This redeploy method is referred to as a "fast redeploy."

> **Note**
>
> If your cartridge has pending or completed orders that you do not want to purge, do not undeploy the cartridge, but deploy the new version with PURGE_CARTRIDGE_BEFORE_DEPLOY set to **false**.

- Set ENTITY_CONFLICT_ACTION_ON_DEPLOY to **replace** to replace the old entities with the new (this is the default); set it to **ignore** to add the new entities and retain the old; or set it to **abort** to stop the process. This variable applies only if PURGE_CARTRIDGE_BEFORE_DEPLOY is set to **false**.
- Set PURGE_ORDER_ON_UNDEPLOY to **true** to purge all existing orders associated with the cartridge, or set it to **false** if you do not want the system to undeploy the cartridge if it has pending orders. If both PURGE_ORDER_ON_UNDEPLOY and FAST_CARTRIDGE_UNDEPLOY are set to **true**, the operation uses the forced fast undeploy option, which quits open orders; neither the cartridge nor the associated orders are purged. If PURGE_ORDER_ON_UNDEPLOY is set to true and FAST_CARTRIDGE_UNDEPLOY is set to false, the operation uses the forced undeploy option, which purges the cartridge and associated orders.

> **Caution**
>
> Undeploying a cartridge purges all existing orders for that cartridge.

5. In Design Studio, in the **customAutomation** directory of your cartridge workspace, replace **automationMap.xsd** with the latest version from the **`$OSM_SDK/Automation/automationdeploy_bin`** directory. Refer to the discussion about defining OSM preferences in the Design Studio Modeling OSM Processes Help for more information about how to define general preferences for the new OSM version.
6. If there are **no pending** Order-to-Activate orders in your OSM instance, change the target version and cartridge versions to be compatible with OSM 8.0. If there are pending Order-to-Activate orders in your OSM instance, skip this step.

   For each cartridge in your workspace, in the Project editor Properties tab set the target version to 8.0.0.0.0 and update the version number if your cartridge is not using five-digit version numbers. See "Updating Cartridges to a Five-Digit Version."
7. Clean and build all cartridges.

   In this step, it is expected that builds will fail. Go to the Problems view and run the quick fix for the following errors:

   - "Automation Build Error - *CartridgeName* has an automation build file (automationBuild.xml). Automation build files are not supported for target version 8.0.0.0.0".

     After fixing the problem, the **automationBuild.xml** file is removed from the **src** directory of the cartridge.
   - "Order Model Error - Order Template Node/ControlData/... is not defined...".

     After fixing the problem, Design Studio upgrades the OracleComms_OSM_CommonDataDictionary model project to the current version for 8.0 and applies the calculatedStartDate data structure from the OSM common data dictionary to the following entities:

     – Order Specifications in the **Order Template** tab

  - Order Components in the **Order Template** tab (if the Order Component contains control data)
  - Create Task in the **Task Data** tab (if the Creation Task contains control data)
- "Order Model Error - Order Template Node/ControlData/Functions/.../duration is not defined...".

  After fixing the problem, Design Studio applies the duration data structure from the OSM common data dictionary to the following entities:

  - Order Specifications in the **Order Template** tab
  - Order Components in the **Order Template** tab (if the Order Component contains control data)
  - Create Task in the **Task Data** tab (if the Creation Task contains control data)

  This step will automatically update the OracleComms_OSM_CommonDataDictionary, as mentioned previously in this step. You can also update the OracleComms_OSM_CommonDataDictionary manually. See "Updating the Common Data Dictionary Manually" for more information.

8. Some cartridges may fail to build if they contain Java classes which implement interfaces that have changed in OSM 8.0.

   To fix this error, open the failed Java files in Design Studio, right-click the error marker, and run the quick fix "add unimplemented methods".

9. If the same Java class exists in different cartridges (for example, if both cartridge A and B define a Java class `com.mycompany.cartridges.log.LogActivity`), then this class must be the same in both cartridges.

10. Clean and build all cartridges in the workspace again.

11. Redeploy all cartridges to the OSM 8.0 run-time environment.

Upgrade Impacts on Cartridges from Previous Releases to OSM 8.0

This section provides information on upgrade impacts to cartridges from previous releases of OSM to version 8.0.

Updating the Common Data Dictionary Manually

If you follow the procedures in "Upgrading Pre-7.3.5 Cartridges to OSM 8.0," it should not be necessary to update the OracleComms_OSM_CommonDataDictionary manually.

Many releases of OSM, including OSM 7.4, contain additions that have been made to the OracleComms_OSM_CommonDataDictionary.

Oracle recommends that you update the data dictionary whenever you upgrade Design Studio. However, if you do not plan to use the new OSM features, it is not mandatory to update it.

All changes made to the common data dictionary are additive and fully backward-compatible. If you decide not to update the data dictionary at the time you update your cartridges, and later you make any changes to your solution that require the updated common data dictionary, you can also update the common data dictionary using the Quick Fix option on the problem marker that informs you that a needed data element is not defined. If you prefer, you can update it using the manual procedure in this section.

To update the common data dictionary manually:

1. In Design Studio, from the **Project** menu, deselect **Build Automatically**.
2. In the Studio Projects view, right-click the **OracleComms_OSM_CommonDataDictionary** project and select **Delete**.
3. In the confirmation dialog box, select **Also delete contents...** and click **Yes**.
4. Create the common data dictionary in your workspace. See "Creating the Common Data Dictionary Project in Your Workspace" for instructions.
5. Clean and build all cartridges in the workspace.
6. If desired, from the **Project** menu, select **Build Automatically**.

Upgrading Service Actions with Explicit Data Elements

Starting with Design Studio 7.3.4, if you have a Service Action with explicit data elements (data elements that are not grayed out in the Action editor **Data Elements** tab), those data elements will only be available in the mapping rule editor if the action has at least one action code and the explicit data elements are each assigned a data direction. (A Service Action is an action with an Action Type of **Service** in the Action editor **Properties** tab.) To add an action code to an Action:

1. In your Design Studio workspace, open the editor for the Action.
2. In the **Action Codes** tab, click **Add**.
3. In the resulting window, click **Select**.
4. Select an action code from the list and click **OK**.
5. Click **OK** again to add the action code.

To assign a data direction for a data element:

1. In your Design Studio workspace, open the editor for the Action.
2. In the **Data Map** tab, click the field corresponding to the action code you have added and data element that you would like to have available in the mapping rule.
3. Select a direction (for example **Optional In**) for the data element.
4. Repeat steps 2 and 3 for each data element you would like to use in the mapping rule editor.

After you have made the changes above, you should save your Action and perform a clean build of your cartridges.

Modeling Order Components to Use Calculated Start Dates

Starting with OSM 7.2.2, you can model order components to use a calculated start date calculated by OSM. OSM derives the start date while balancing order item durations and dependencies and the order component requested delivery date.

This functionality depends on two data structures existing in the OSM common data dictionary: calculatedStartDate and duration.

See "Upgrading OSM to 8.0" for more information on adding these data structures to the OSM common data dictionary.

After migrating a cartridge to version 8.0, you must set the cartridge target version to OSM version 8.0. If you set the target version to OSM version 7.2 or earlier, OSM does not calculate

start dates for order components. Instead, order components begin processing as soon as possible, based on dependencies.

See *OSM Modeling Guide* for guidelines about modeling order components to use calculated start dates.

Turning On Inheritance of Keys and Significance for Existing Cartridges

Starting with OSM 7.2, significance and keys are included in the information extended from a base order. The significance of an inherited data element within the order template is now inherited from the OSM entity that contributed it. For example, a data element an order component contributes to the **ControlData** structure defined on the order now inherits the significance value that is defined on the order component.

Providing this inheritance within the order template is recommended for new cartridge development. You have the option to turn this functionality on for existing cartridges.

Design Studio will detect conflicts such as incorrect significance behavior after you turn the functionality on, and raise problem markers. To resolve conflicts you must examine order data elements and align their significance values to adhere to inheritance rules (as described below). You must resolve problem markers before you can deploy your cartridges.

The order template inheritance functionality may impact cartridges that were developed in earlier releases of Design Studio as follows:

- Unintended inheritance

  A data element now inherits its significance value from its order contributor; for example, from a base order, an order component, or an order item specification. If you intended for a data element marked **Inherited** in significance to inherit its significance value from its data schema, you must set this manually.

  To ensure the inheritance uses the significance value you intended, check all data elements and explicitly mark them as significant or not according to your design requirements.

- Inheritance discrepancies due to multiple order template contributors

  In the case of multiple inheritance where a data element is inherited on the order template from multiple entities, the inherited data element must have the same significance value as is set for the OSM entity from which it is inherited. For example, the **ControlData/Functions** data structure can be inherited on the order template from these OSM entities: an extended order, an order component, and an order item. If the **ControlData/Functions** data structure is set with a different significance value in any of the contributing OSM entities (on the **Order Template** tab of those entities), Design Studio creates a problem marker. You cannot build and deploy the cartridge until the marker is resolved.

  > **Note**
  >
  > Design Studio creates problem markers only if the order contributors have different values and the order has the significance set to **Inherited**.

  To resolve problem markers, examine the relevant data elements and align the significance so that it is consistent across all order contributors. You can set the same significance value on all Order Component Specification editor **Order Template** tabs for all order components contributing to the order, or you can override the significance value of that data structure on the order by using the Order editor.

After you resolve design discrepancies that may exist, use OSM preferences to turn on order template inheritance of keys and significance.

To turn on order template inheritance of keys and significance for existing cartridges:

> **Tip**
>
> Disallow this functionality *temporarily* to minimize the immediate upgrade impact on your cartridges, and allow it later for optimal development convenience.

1. In Design Studio, from the **Window** menu, select **Preferences**, then select **Oracle Design Studio**, and then select **Order and Service Management Preferences**.
2. In the **Order Template Inheritance** field, do the following:
   - To allow significance defined on an order template data element to be inherited from child orders and other order contributors, select **Inherit significance from order contributors**.
   - To allow keys defined on an order to be inherited from child orders and from other order contributors, select **Inherit keys from order contributors**.

   Design Studio sets the order template inheritance preferences you specify as a global preference.
3. Click **OK**.

   Design Studio saves your order template inheritance preferences and closes the Preferences dialog box.
4. Clean and build the cartridges.
5. Resolve any problem markers that may result from order template inheritance discrepancies in your cartridges.

Creating the Common Data Dictionary Project in Your Workspace

In releases of Design Studio prior to 7.2, you were required to manually model order component control data. Starting with release 7.2, Design Studio automatically creates control data structures for order components if you create the **OracleComms_OSM_CommonDataDictionary** model project in your workspace.

The common data dictionary project can coexist with existing cartridge projects and will automatically generate order component control data for new order components you create after creating the model project in your existing workspace. In this case, Design Studio updates the order component control data structure you have defined in the data schema of your existing base project. However, Design Studio does not replace the reference of **ControlData** in the order templates to use the model project schema instead of your existing project schema.

Design Studio will prompt you to create the **OracleComms_OSM_CommonDataDictionary** cartridge the first time you create or open an entity related to orchestration in a workspace using Design Studio 7.2 or later where the common data dictionary is not present.

If in the past you have seen this prompt and selected **Do not show this prompt in the future**, you must re-enable the prompt in order to create the common data dictionary.

To create the common data dictionary in your workspace:

1. If you have previously disabled the prompt:

a. From the **Window** menu, select **Preferences**.

   The Preferences dialog box is displayed.

b. In the Preferences navigation tree, expand **Oracle Design Studio**, and then expand **Order and Service Management Preferences**.

c. Select **Orchestration Preferences**.

d. In the Common Data Dictionary area, select the **Prompt To Create Orchestration Data Dictionary** check box.

e. Click **OK**.

   Design Studio saves the preference and closes the Orchestration Preferences dialog box.

2. Open or create an orchestration entity, for example, an order item specification or an order component specification.

   You are prompted to confirm importing the common data dictionary.

3. Click **OK**.

   The **OracleComms_OSM_CommonDataDictionary** model project is created in your workspace.

See Design Studio Help for information on how the **OracleComms_OSM_CommonDataDictionary** model project creates control data.

Handling Three-Digit and Five-Digit Cartridge Version Numbers

Prior to OSM 7.2, cartridges used three digits to specify their version. Starting with OSM 7.2, cartridges use five digits to specify their version. If you choose to change the target version of your cartridges for new orders, you must update the version format.

Processing In-Flight Orders That Use a Three-Digit Version

If you have open orders that use the three-digit format, the cartridges must continue to use the three-digit format until the orders are completed. You may want to have the existing cartridges retain their three-digit version numbers, and create new versions of the cartridges that have the updated five-digit version numbers. For information about cartridge versioning, see *OSM Modeling Guide*. For instructions on changing the number of digits in the version number, see, "Updating Cartridges to a Five-Digit Version." If you are upgrading cartridges from an OSM version prior to 7.0.3, you should build the original version of the cartridges in Design Studio with a target version of 7.0.3 to update the cartridges as far as possible without changing the length of the version numbers.

Updating Cartridges to a Five-Digit Version

Starting with OSM 7.2, cartridges use five digits to specify their version. If you choose to change the target version of your cartridges, you must update the version format.

To update cartridges to use a five-digit version number:

1. In the Project editor Properties tab for each cartridge, change the value of **Target Version** from the current value (7.0.3 or earlier) to **7.4.0**.

2. Update the cartridge versions to 1.0.0.0.0 in the XML catalogs.

   a. Open the Package Explorer view.

b. For each cartridge, open the **catalog.xml** file located in *CartridgeName*/**xmlCatalogs/core**.

c. Select the **Source** tab.

d. Each rewritePrefix attribute in the file includes the version 1.0.0. Change this version to 1.0.0.0.0.

3. Set up the event views for the orders in the cartridge.
4. Exit and restart Eclipse. If prompted to save any entities, select **Yes**.
5. When Eclipse is restarted, the Studio Project Upgrade window is displayed with a list of cartridges that should be upgraded. Click **Finish** to upgrade the cartridges. This can take several minutes. When the upgrade is finished, you are presented with the option to view the upgrade logs if you wish.
6. Build all of the cartridges. Some cartridge builds will fail with the following error listed in the Problems pane:

```
Automation Build Error - {Cartridge Name} has an automation build file
(automationBuild.xml). Automation build files are not supported for target version
8.0. Right click on problem marker for the Quick Fix.
```

For each cartridge that fails to build with this error:

a. Right-click the error in the Problems pane and select **Quick Fix** from the menu.

b. Click **Finish** in the Quick Fix window and then click **OK** to confirm.

c. Rebuild the project.

Specifying Task Views for Order-Related Automation

Starting with OSM 7.2, you can use the Automation View field to specify the query view that the automation plug-in uses for Order, Event and Jeopardy notifications. One default query task is available per role. You can add roles and configure default query tasks in the Order editor Permissions tab. In addition to triggering automation plug-ins, order data changed event notifications can trigger one other type of notification. See the discussion on defining event notifications in *OSM Modeling Guide* for more information.

This does not apply to Automation Tasks because data available to the automation plug-in is explicitly defined as a Task View.

Configuring Order Lifecycle Policy Transition Error Messages

Starting with OSM 7.2 you can configure the error messages and severity levels logged when an Order Lifecycle Policy condition fails instead of the full XQuery code being logged. For more information, see the discussion on the Order Lifecycle Policy Transition Conditions tab in the *Design Studio Modeling OSM Processes Help*.

When cartridges are migrated, each Order Lifecycle Policy condition is set with the default level of ERROR. The first 1000 characters of the condition expression are the initial error message. After migration, you should re-configure the error message and level and the condition name in Design Studio.

Modeling Data Entries Above the 1000-Character Limit

Starting with Design Studio 3.1.4 (compatible with OSM versions up to 7.0.3), Design Studio generates an error message to prevent creation of data entries above 1000 characters. In previous versions of Design Studio, you could create unbounded data dictionary entries (up to

9999 characters) in OSM cartridges and deploy the cartridges to OSM servers (up to OSM release 7.0.3). These OSM cartridge data entries were persisted in the OSM database as metadata that OSM used for creating run-time entities associated with orders.

However, the OSM database has always had a data entry limit of 1000 characters. Cartridges deployed with unbounded data entries using Design Studio 3.1.3 to an OSM 7.0.3 server would generate no errors, but at run time, if an order was submitted with a data entry of more than 1000 characters, the order processing would fail. This failure was caused by the OSM database data entry limit which does not normally support unbounded lengths greater than 1000 characters.

Following is an example of the Design Studio error messages generated in this situation:

```
Data Dictionary Model Error – Max length of the Data Element Bandwidth backing the Order
Template node/adsl_service_details/bandwidth exceeds 1000.
```

You must resolve this error before you can deploy the OSM cartridge.

If there is a requirement for data entries that contain more than 1000 characters, OSM provides other ways to achieve this goal including:

- Modeling data dictionary entries as XML type so that the data entry can contain XML documents.
- Adding order remarks
- Adding attachments to order remarks

See *OSM Modeling Guide* for more information about these alternatives.

> **Note**
>
> Although OSM provides powerful orchestration and order provisioning engines, OSM is not a data warehouse and management system and should not be used as an intermediary between upstream and downstream systems.

12

Updating Order-to-Activate Cartridges

The Order-to-Activate cartridges are pre-built Oracle Communications Order and Service Management (OSM) cartridges that support the Oracle Order-to-Activate business process to be used with the Oracle Communications Order to Cash Integration Pack for Oracle Communications Order and Service Management (Order to Cash Integration Pack for OSM). For more information about this product, see *OSM Cartridge Guide for Oracle Application Integration Architecture*.

This chapter describes how to update earlier versions of OSM Order-to-Activate cartridges to run on OSM 8.0. It also contains information about how to configure Oracle WebLogic Server resources for the cartridges in your WebLogic Server domain for OSM 8,0, and how to deploy the updated cartridges.

System Requirements for Updating Order-to-Activate Cartridges

Before updating the Order-to-Activate cartridges, ensure the following:

- OSM is upgraded to 8.0.
- Oracle Communications Service Catalog and Design - Design Studio is upgraded to a version compatible with OSM 8.0 and your workspace is configured accordingly. See *Service Catalog and Design Compatibility Matrix* for details.

  The instructions in this chapter assume you are using a new (empty) Design Studio workspace for OSM.
- The Oracle WebLogic Server domain for OSM 8.0 is running.
- Existing user-created domain configurations, such as JMS queues, users, groups, and emulators are the same as before the OSM upgrade. (Do not re-create the system-generated users: osm, osmoe, osmde, osmfallout, osmlf, osmoelf, osmlfaop, and uim.)

Updating the Order-to-Activate 2.1.1 Cartridges

Because the Order-to-Activate 2.1.1 cartridges support Asset Management functionality, which is not supported in OSM 8.0, these cartridges can only be used with OSM 7.3.1 and cannot be updated to work with OSM 8.0.

Preparing to Update the Order-to-Activate 7.2, 2.0.1, 2.1.0, and 2.1.2 Cartridges

Before updating Order-to-Activate cartridges, do the following:

- Ensure compatibility between Order-to-Activate cartridges, OSM, and Oracle AIA. See "Ensuring Order-to-Activate Cartridge Compatibility."
- Back up your existing cartridges.
- Ensure that you have the latest patch for your version of the cartridges. See "Getting the Latest Patch for Your Version of the Cartridges."

- Start the Oracle WebLogic Server domain for OSM 8.0.
- Set Design Studio preferences. See "Setting Design Studio Preferences."
- Download the migration package. See "Downloading the Migration Package."
- Import the Migration Package Cartridge. See "Importing the Migration Package Cartridge."

Ensuring Order-to-Activate Cartridge Compatibility

To install or upgrade the Order-to-Activate cartridges, you must ensure compatibility between the following:

- OSM software version and Order-to-Activate cartridge version

  OSM is compatible with all cartridges developed in a previous release, including Order-to-Activate cartridges. So, any OSM version is compatible with the same version or an earlier version of the Order-to-Activate cartridges.

- OSM Order-to-Activate cartridge version and Oracle Application Integration Architecture (Oracle AIA) Order to Cash Integration Pack for OSM version

For Order-to-Activate cartridge compatibility information see *Order-to-Activate Cartridge Product Compatibility Matrix* (in the **OSM Cartridges for Oracle Application Integration Architecture** section of the OSM documentation) on the Oracle Help Center website:

http://docs.oracle.com/en/industries/communications/order-service-management/index.html

> **Note**
>
> If you want to use Order-to-Activate 7.2.0.x cartridges with OSM 8.0, you must ensure that your Order-to-Activate cartridges have been upgraded to the 7.2.0.2 patch or higher.

Getting the Latest Patch for Your Version of the Cartridges

It is important to get the latest patch for the version of the cartridge you are updating. For the 7.2 version of the cartridges, you can find the latest patch on Oracle support inside the latest OSM patch for the appropriate version. For the 2.0.1 and later versions of the cartridges, the latest patch is on Oracle support as a separate release for Oracle AIA cartridges under the OSM product. Patches contain a complete installation of the cartridges and do not require an existing installation to be present. It is important to use the latest patch regardless of whether you have modified the Order-to-Activate solution.

Setting Design Studio Preferences

To set Design Studio preferences:

1. Verify that Design Studio is running.
2. From the **Window** menu, select **Preferences**.

   The Preferences dialog box is displayed.
3. In the Preferences navigation tree, expand **Oracle Design Studio**.
4. Select **Order and Service Management Preferences.**

The Order and Service Management Preferences page includes the Deploy Properties pane in which you can provide home directories for various tools.

5. In the **WebLogic Home** field, enter or browse to the directory in which the WebLogic Server is installed, for example, **C:\Program Files\Oracle Middleware\wlserver**.
6. In the **Java SDK Home** field, enter or browse to the directory in which you have installed the JDK for the version of Java that matches the version of Java on your OSM server. See *OSM Compatibility Matrix* for details about the required version.
7. In the **OSM SDK Home** field, enter or browse to the directory in which you have installed the OSM SDK, for example, **C:\Program Files\Oracle Communications\OSM7\SDK**.
8. Select **Inherit significance from order contributors** and **Inherit keys from order contributors**.
9. Expand **Order and Service Management Preferences** and select **Application Integration Architecture (AIA) Preferences.**
10. In the **Oracle Middleware Home** field, enter the directory in which you have installed Oracle Middleware products, for example, **C:\Program Files\Oracle Middleware**.
11. Under **Java**, select **Installed JREs**.
12. If the Java directory that you entered for **Java SDK Home** in step 6 is not displayed, add it and ensure that it is selected.
13. Click **OK**.
14. From the **Project** menu, deselect **Build Automatically**.

Downloading the Migration Package

The Migration package is included with the OSM 8.0 software download, and does not need to be downloaded separately. Your existing OSM software download should include the following file:

OracleComms_OSM_O2A_CartridgesMigration.zip

This is the migration package.

Updating the Order-to-Activate Cartridges

You can update the Order-to-Activate Cartridges to run on OSM 8.0 in any of the following ways:

- Updating the Order-to-Activate Cartridges By Using Migration Scripts
- Updating Order-to-Activate Cartridges Manually

Updating the Order-to-Activate Cartridges By Using Migration Scripts

Before you update the Order-to-Activate cartridges, you must import the migration package cartridge.

Importing the Migration Package Cartridge

To import the migration package cartridge:

1. Verify that Design Studio is running.
2. From the **Studio** menu, select **Show Design Perspective**.

3. From the **Window** menu, select **Show View**, and then select **Package Explorer**.
4. From the **Window** menu, select **Show View**, and then select **Other**.

   The Show View window is displayed.
5. Expand **Ant** and click **Ant** from below it. Click **OK**.

   The Ant view opens.
6. From the **File** menu, select **Import Studio Project**.

   The Import Projects dialog box is displayed.
7. Select **Select archive file** and click **Browse**.
8. Browse to the directory where you extracted the OSM 7.5 software and select **OracleComms_OSM_O2A_CartridgesMigration.zip**.
9. Click **Open**.

   OSM.O2A.Cartridges.Migration is displayed and selected in the **Projects** field.
10. Click **Finish**.

    The **OSM.O2A.Cartridges.Migration** project is imported.

The **OSM.O2A.Cartridges.Migration** project contains the following directories, which you can see in the Package Explorer view:

- **7.2.0**: This directory contains the resources required for migrating Order-to-Activate 7.2 cartridges to run on OSM 7.5.
- **2.0.1.3**: This directory contains the resources required for migrating Order-to-Activate 2.0.1.3 cartridges to run on OSM 7.5.
- **2.0.1.4**: This directory contains the resources required for migrating Order-to-Activate 2.0.1.4 cartridges to run on OSM 7.5.
- **2.1.0.1**: This directory contains the resources required for migrating Order-to-Activate 2.1.0.1. cartridges to run on OSM 7.5.
- **2.1.0.2**: This directory contains the resources required for migrating Order-to-Activate 2.1.0.2. cartridges to run on OSM 7.5.
- **2.1.2.0**: This directory contains the resources required for migrating Order-to-Activate 2.1.2.0. cartridges to run on OSM 7.5.

Updating Unmodified or Modified Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges to Run on OSM 7.5

There are four distinct scenarios for updating the Order-to-Activate cartridges, depending on your needs. Each scenario has its own process, outlined below.

- To update Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x cartridges, see the following sections:
  - Importing the Installation Cartridge for Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x
  - Importing Unmodified OSM Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges

    or

    Importing Modified Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges
  - Migrating the Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges

  - Configuring WebLogic Server Resources for Order-to-Activate 2.0.1.x, 2.1.0.x, or 2.1.2.x
- To update Order-to-Activate 7.2 cartridges, see the following sections:
  - Importing the Installation Cartridge for Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x
  - Importing Unmodified OSM Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges

    or

    Importing Modified Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges
  - Migrating the Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges
  - Configuring WebLogic Server Resources for Order-to-Activate 7.2

> **Note**
>
> Some XQuery functions that were written for earlier versions of Saxon might not work correctly. For more information, see "XQuery Model Changes."

The following procedure updates Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x cartridges so that they can run on OSM 8.0. Use this procedure both to update the standard, unmodified Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x or 2.1.2.x cartridges and to update a cartridge that is based on modified versions of the Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x cartridges. This procedure makes no functional changes to the contents of the cartridges.

Importing the Installation Cartridge for Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x

To import the installation cartridge for Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x:

1. Ensure that you have the following software installed on your Windows system:
   - The supported version of WebLogic Server and ADF. See "Software Requirements" for more information.
   - OSM Administrator Software Development Kit (SDK) components
   - Java JDK: Use the version of Java that matches the one being used by the OSM server. See "Software Requirements" for more information.
   - Eclipse with Design Studio plug-ins

   > **Note**
   >
   > The Order-to-Activate cartridges require the following Design Studio plug-ins:
   >
   > - Design Studio Platform
   > - Design Studio for Order and Service Management
   > - Design Studio for Order and Service Management Orchestration
   > - Design Studio for Order and Service Management Integration
   > - Design Studio for Order and Service Management Orchestration Application Integration Architecture (AIA)
   >
   > See *Design Studio Installation Guide* for information about installing Design Studio plug-ins and how to confirm which plug-ins are installed.

2. If you do not already have it available, locate or download the appropriate version of the unmodified cartridges.
   - You should always ensure that you use the latest patch of the cartridges, if one is available. See "Getting the Latest Patch for Your Version of the Cartridges" for more information.
   - If you are downloading the latest version from the Oracle software delivery website, select **Oracle Communications Order and Service Management Cartridge for Provisioning Fulfillment** and select your platform.
   - If you have already downloaded the Order-to-Activate cartridges, locate the **OracleComms_OSM_O2A_CartridgesInstaller_b*yyyymmdd*.zip** file.
3. Unzip the **OracleComms_OSM_O2A_CartridgesInstaller_b*yyyymmdd*.zip** file.

   The **OSM.PIP** directory containing the **OracleComms_OSM_O2A_Install.zip** file is created.
4. Verify that Design Studio is running.
5. From the **File** menu, select **Import Studio Project**.

   The Import Projects dialog box is displayed.
6. Select **Select archive file** and click **Browse**.
7. Browse to the **OSM.PIP** directory and select **OracleComms_OSM_O2A_Install.zip**.
8. Click **Open**.

   The **OracleComms_OSM_O2A_Install** project is displayed and selected in the **Projects** field.
9. Click **Finish**.

   The **OracleComms_OSM_O2A_Install** project is imported.

Importing Unmodified OSM Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges

If you are updating unmodified Order-to-Activate cartridges, follow this procedure.

To import the OSM Order-to-Activate cartridges:

1. Open the Ant view.
2. Right-click in the Ant view and select **Add Buildfiles**.

   The Buildfile Selection dialog box is displayed.
3. Expand **OracleComms_OSM_O2A_Install** and select **OSM.O2A.Installation.xml**.
4. Click **OK**.

   The **OSM.O2A.Installation** item is displayed in the Ant view.
5. Right-click **OSM.O2A.Installation** and select **Run As**.
6. Select **Ant Build...**, (not **Ant Build**).

   The Edit Configuration dialog box is displayed.
7. Click the **Build** tab and deselect **Build before launch**.
8. Click the **Properties** tab and deselect **Use global properties as specified in the Ant runtime preferences**.
9. Click the **JRE** tab and select **Run in the same JRE as the Workspace**.
10. Click **Close** and click **Yes**.

11. In the Ant view, expand **OSM.O2A.Installation** and double-click **import_solution**.
12. In the first **Ant Input Request** window, do one of the following:
    - To import cartridges for the Complex solution topology, enter **c** and click **OK**.
    - To import the cartridges for the Typical solution topology, enter **t** and click **OK**.
    - To import the cartridges for the Simple solution topology, enter **s** and click **OK**.
13. In the second **Ant Input Request** window, do one of the following:
    - To deploy both central order management and service order management to the same OSM instance, enter **s** and click **OK**.
    - To deploy central order management and service order management to different OSM instances, enter **m** and click **OK**.

    The cartridges appropriate for the settings you selected are imported into the workspace. This may take a few minutes.

Importing Modified Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges

If you are updating modified Order-to-Activate cartridges, follow this procedure.

To import the modified OSM Order-to-Activate cartridges:

1. Import the latest patch of the Order-to-Activate cartridges for your version into your Design Studio workspace for OSM 8.0.

   For example, if you have been working with Order-to-Activate version 7.2.0.6, import the cartridges for the latest patch of 7.2.0 (for example, 7.2.0.10).

   For more information about how to do this see "Importing Unmodified OSM Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges."
2. Import any custom cartridges that you created into the workspace.
3. If you have modified any sample Order-to-Activate cartridges (which are unsealed), such as composite cartridges, check the README file for the latest patch, which tells you what cartridges have been changed by the patch. Then do one of the following:
   - If the cartridges you modified have not been updated by the patch, import your modified versions of the cartridges.
   - If the cartridges you modified have been updated by the patch, you must re-create your modifications in the versions of the cartridges from the latest patch.
4. In general, you should not unseal and modify cartridges that are sealed. If you have done this, check the README file for the latest patch, which tells you what cartridges have been changed by the patch. Then do one of the following:
   - If the cartridges you modified have not been updated by the patch, import your modified versions of the cartridges.
   - If the cartridges you modified have been updated by the patch, you must re-create your modifications in the versions of the cartridges from the latest patch.

Migrating the Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x Cartridges

To migrate the modified or unmodified Order-to-Activate 7.2, 2.0.1.x, 2.1.0.x, or 2.1.2.x cartridges:

1. Right-click in the Ant view and select **Add Buildfiles**.

   The Buildfile Selection dialog box is displayed.

2. Expand the **OSM.O2A.Cartridges.Migration** cartridge and expand the **7.2.0**, **2.0.1.3**, **2.0.1.4**, **2.1.0.1**, **2.1.0.2**, or **2.1.2.0** directory, depending on your version of the cartridges. Click the **migration.xml** file and click **OK**.

   The Ant view displays the appropriate file from the files listed below:

   - **OSM.O2A.Cartridges.7.2.0_to_7.4_Migration**
   - **OSM.O2A.Cartridges.2.0.1.3_to_7.4_Migration**
   - **OSM.O2A.Cartridges.2.0.1.4_to_7.4_Migration**
   - **OSM.O2A.Cartridges.2.1.0.1_to_7.4_Migration**
   - **OSM.O2A.Cartridges.2.1.0.2_to_7.4_Migration**
   - **OSM.O2A.Cartridges.2.1.2.0_to_7.4_Migration**

3. Configure the buildfile for the file you have added:

   a. In the Ant view, right-click the name of the file and select **Run As**.

   b. Select **Ant Build...** (not **Ant Build**).

      The Edit Configuration dialog box is displayed.

   c. Click the **Build** tab and deselect **Build before launch**.

   d. Click the **Properties** tab and deselect **Use global properties as specified in the Ant runtime preferences**.

   e. Click the **JRE** tab and select **Run in the same JRE as the Workspace**.

   f. Click **Close** and click **Yes**.

4. In the Ant View, expand the file you added (**OSM.O2A.Cartridges.7.2.0_to_7.4_Migration**, **OSM.O2A.Cartridges.2.0.1.3_to_7.4_Migration**, **OSM.O2A.Cartridges.2.0.1.4_to_7.4_Migration**, **OSM.O2A.Cartridges.2.1.0.1_to_7.4_Migration**, **OSM.O2A.Cartridges.2.1.0.2_to_7.4_Migration** or **OSM.O2A.Cartridges.2.1.2.0_to_7.4_Migration**) and double-click **migrate**.

5. Exit and restart Eclipse.

   If prompted to save any entities, select **Yes**.

   When Eclipse is restarted, the Studio Project Upgrade window is displayed, with a list of cartridges that should be upgraded.

6. Click **Finish**.

   The cartridges will be updated, including changing the data structures in the data dictionary. This can take several minutes. When the upgrade is finished, you are presented with the option to view the upgrade logs.

7. Exit and restart Eclipse.

   If prompted to save any entities, select **Yes**.

8. Build all of the cartridges.

   See the Design Studio Help for information about how to build cartridges. Some cartridge builds will fail with the following error listed in the Problems pane:

   ```
   Naming Conflict - Entity names must be unique (case-insensitively) amongst other entities of the same or similar types within a workspace.  The conflicts are: /OracleComms_Model_Base/model/Activation.applicationRole.
   ```

9. To solve the problem listed above, do the following:

a. Open the Package Explorer view.

b. Navigate to the **OracleComms_Model_BaseCatalog/model** directory.

c. Remove all files in this directory that have the **.applicationRole** extension.

10. Clean and build all of the cartridges.

11. Configure the WebLogic Server resources for your environment.

    See "Configuring WebLogic Server Resources for Order-to-Activate 7.2" for more information.

Configuring WebLogic Server Resources for Order-to-Activate 2.0.1.x, 2.1.0.x, or 2.1.2.x

The process in this section configures the metadata for the composite cartridges in addition to configuring the WebLogic Server resources.

To configure the WebLogic Server resources for Order-to-activate 2.0.1.x, 2.1.0.x, or 2.1.2.x see *OSM Cartridge Guide for Oracle Application Integration Architecture* for the version of the cartridges you are updating. In the chapter titled "Performing an Interactive Installation of the Order-to-Activate Components" (version 2.1.0 or later) or "Installing the Order-to-Activate Components" (version 2.0.1), and perform the procedure in the "Configuring WebLogic Server Resources" section.

This documentation is available from the Oracle Help Center website:

http://docs.oracle.com

Configuring WebLogic Server Resources for Order-to-Activate 7.2

The process in this section configures the metadata for the composite cartridges in addition to configuring the WebLogic Server resources.

To configure the WebLogic Server resources for Order-to-Activate 7.2:

1. Open the Ant view.

2. Find the one row in Table 12-1 that matches your situation. For each cartridge listed in the corresponding "SolutionConfig.xml Files to Add" column of the table:

   a. Right-click in the Ant view and select **Add Buildfiles**.

      The Buildfile Selection dialog box is displayed.

   b. Expand the cartridge listed in the table and click on the **SolutionConfig.xml** file.

   c. Click **OK**.

Table 12-1 SolutionConfig.xml Files to Use in Different Situations

Topology	Configuration	SolutionConfig.xml Files to Add
Simple	Central order management and service order management are on **the same** OSM instance	OracleComms_OSM_O2A_COMSOM_SimpleSolution
Simple	Central order management and service order management are on **different** OSM instances	OracleComms_OSM_O2A_COM_SimpleSolution OracleComms_OSM_O2A_SOM_Solution
Typical	Central order management and service order management are on **the same** OSM instance	OracleComms_OSM_O2A_COMSOM_TypicalSolution

Table 12-1 (Cont.) SolutionConfig.xml Files to Use in Different Situations

Topology	Configuration	SolutionConfig.xml Files to Add
Typical	Central order management and service order management are on **different** OSM instances	OracleComms_OSM_O2A_COM_TypicalSolution OracleComms_OSM_O2A_SOM_Solution

The items are displayed in the Ant view. Each **SolutionConfig.xml** file is listed as the name of the cartridge it was added from. For example, if you added the **SolutionConfig.xml** file from **OracleComms_OSM_O2A_COMSOM_SimpleSolution**, it is listed as **OracleComms_OSM_O2A_COMSOM_SimpleSolution** in the Ant view.

3. For each **SolutionConfig.xml** file you have added, configure the buildfile:
   a. In the Ant view, right-click the name of the cartridge for the **SolutionConfig.xml** file and select **Run As**.
   b. Select **Ant Build...** (not **Ant Build**).

      The Edit Configuration dialog box is displayed.
   c. Click the **Build** tab and deselect **Build before launch**.
   d. Click the **Properties** tab and deselect **Use global properties as specified in the Ant runtime preferences**.
   e. Click the **JRE** tab and select **Run in the same JRE as the Workspace**.
   f. Click **Close** and click **Yes**.
   g. Right-click the name of the cartridge for the **SolutionConfig.xml** file and select **Run As** again.
   h. Select **Ant Build...** (not **Ant Build**).

      The Edit Configuration dialog box is displayed.

      > **Note**
      >
      > It is necessary to close and reopen the Edit Configuration dialog box because after you have deselected the **Use global properties...** check box, Eclipse prevents you from changing any of these properties until you close and re-open the Edit Configuration dialog box.

   i. Click the **Properties** tab and set the appropriate values according to Table 12-2.

Table 12-2 Values for Ant Edit Configuration Properties Tab

Property Name	Description	Notes
aia.emulator.serverName	Name of the cluster or server within WebLogic Server to which you want to deploy the emulators. Set this to one of the following: • If OSM is installed to a cluster, set this value to the name of the cluster. • If OSM is installed to the administration server, set this to the name of the administration server. • If OSM is installed to a single managed server, set this value to the managed server name. If both central order management and service order management are in the same OSM server instance, set this to the name of the cluster or server for the single OSM instance. If central order management and service order management are in different OSM server instances, set this to the name of the cluster or server for central order management in the central order management buildfile and to the name of the cluster or server for service order management in the service order management buildfile.	Set this property in all files if you are installing the Oracle AIA emulators.
cf.adminServerListenAddress	Host name of the system where the WebLogic Server for central order management is running. If you are in a clustered environment, set this to the server where the Administration server is located.	Set this if the name of the cartridge associated with the buildfile contains **COM** or **COMSOM**.
cf.adminServerListenPort	Port on which the WebLogic Server for central order management is listening. If you are in a clustered environment, set this to the port on which the Administration server is listening.	Set this if the name of the cartridge associated with the buildfile contains **COM** or **COMSOM**.
cf.clusterName	Name of the cluster for central order management, exactly as it is shown in the WebLogic Remote console.	Set this if the name of the cartridge associated with the buildfile contains **COM** or **COMSOM** and you are in a clustered WebLogic environment.
cf.userName	Name of a user with administrative privileges on the WebLogic Server for listening on cf.adminServerListenAddress and cf.adminServerListenPort.	Set this if the name of the cartridge associated with the buildfile contains **COM** or **COMSOM**.
lf.adminServerListenAddress	Host name of the system where the WebLogic Server for service order management is running. If you are in a clustered environment, set this to the server where the Administration server is located.	Set this if the name of the cartridge associated with the buildfile contains **SOM**.
lf.adminServerListenPort	Port on which the WebLogic Server for service order management is listening. If you are in a clustered environment, set this to the port on which the Administration server is listening.	Set this if the name of the cartridge associated with the buildfile contains **SOM**.
lf.clusterName	Name of the cluster for service order management, exactly as it is shown in the WebLogic Remote console.	Set this if the name of the cartridge associated with the buildfile contains **SOM** and you are in a clustered WebLogic environment.

Table 12-2 (Cont.) Values for Ant Edit Configuration Properties Tab

Property Name	Description	Notes
lf.userName	Name of a user with administrative privileges on the WebLogic Server listening on lf.adminServerListenAddress and lf.adminServerListenPort.	Set this if the name of the cartridge associated with the buildfile contains **SOM**.

j. Click **Close** and click **Yes**.

4. For *each* **SolutionConfig.xml** file you have added, do the following:

   a. In the Ant view, expand the cartridge name and double-click **config_All**.

   b. The first Ant Input Request window requests the WebLogic administrator user password. Enter the password for the user you entered in **cf.userName** or **lf.userName** (whichever value you configured for the buildfile you are running). Click **OK**.

   c. In the second Ant Input Request window, enter **y** to use the same password for all of the users being created or enter **n** to use a different password for each user. Click **OK**.

   d. Enter the passwords requested for the Order-to-Activate users by the Ant Input Request windows:

> **Note**
>
> Ensure that the passwords you enter meet the security requirements of your WebLogic Server domain. By default, the WebLogic server requires passwords of at least eight characters, with at least one numeric or special character. However, the requirements for your domain may be different.

   If you entered **y** in the previous step, enter the common password for the Order-to-Activate users and click **OK**.

   If you entered **n** in the previous step and the name of the cartridge associated with the buildfile contains **COM**, you are prompted for passwords for the following users: **COM user** (osm), **COM Order Event user** (osmoe), **COM Data Change Event user** (osmde), and **COM Fallout user** (osmfallout). Click **OK** after each entry.

   If you entered **n** in the previous step and the name of the cartridge associated with the buildfile contains **SOM**, you are prompted for passwords for the following users: **SOM user** (osmlf), **SOM Order Event user** (osmoelf), and **SOM Order Abort user** (osmlfaop). Click **OK** after each entry.

   If you entered **n** in the previous step and the name of the cartridge associated with the buildfile contains **COMSOM**, you are prompted for passwords for the following users: **COM user** (osm), **COM Order Event user** (osmoe), **COM Data Change Event user** (osmde), **COM Fallout user** (osmfallout), **SOM user** (osmlf), **SOM Order Event user** (osmoelf), and **SOM Order Abort user** (osmlfaop). Click **OK** after each entry.

   After you have entered the passwords and clicked **OK**, the system creates the users in the WebLogic domain. This may take a few minutes.

> **Note**
>
> Although **config_All** has now created users in the WebLogic Server domain, it is still possible to cancel the script at a later point and rerun it. If **config_All** finds the users already present in the domain, it will skip adding them again and continue with the rest of the configuration process.

e. In the next Ant Input Request window, enter **s** if you are using a standalone WebLogic Server environment or enter **c** if you are using a clustered environment. Click **OK**.

f. In the next Ant Input Request window, enter **d** (for development environment) if you do not intend to connect to Oracle AIA or enter **p** (for production environment) if you intend to connect to Oracle AIA. Click **OK**.

   If you selected **d**, the queues for a development environment will be created. This may take several minutes. If you selected **p**, the queues will be created after the next step.

g. Do one of the following:

   If you selected to have a development environment, in the next Ant Input Request window, enter **d** to deploy the Oracle AIA emulators or enter **n** to skip deploying the Oracle AIA emulators.

   If you selected to have a production environment, in the next Ant Input Request window, enter **s** and click **OK**. This will install a Store-and-Forward (SAF) service for communications with Oracle AIA. Do not choose **b** in this window, because it is not supported.

h. If you selected to have a production environment, entered **s** to install a SAF service, and the name of the cartridge associated with the buildfile contains **COMSOM** or **COM**:

   Enter the Oracle AIA server user name that OSM central order management should use to connect to Oracle AIA in the next window, and click **OK**.

   Then, enter the password for that user in the next window, and click **OK**.

   Then, enter the host name and port (in the format *hostname*:*port*) that central order management should use for connecting to Oracle AIA in the next window. If the Oracle AIA server is on a WebLogic Cluster, enter all the host names and ports, separated by commas, for example:

   ```
   yourhost1.com:8001,yourhost1.com:8002,yourhost2.com:7030
   ```

   Click **OK**.

   Finally, if OSM is not deployed in a clustered WebLogic Server environment, enter the name of the WebLogic server where central order management is running, and click **OK**.

i. If you selected to have a production environment, entered **s** to install a SAF service, and the name of the cartridge associated with the buildfile contains **SOM**:

   Enter the Oracle AIA server user name that OSM service order management should use to connect to Oracle AIA in the next window, and click **OK**.

   Then, enter the password for that user in the next window, and click **OK**.

   Then, enter the host name and port (in the format *hostname*:*port*) that service order management should use for connecting to Oracle AIA in the next window. If the Oracle AIA server is on a WebLogic Cluster, enter all the host names and ports, separated by commas, for example:

```
yourhost1.com:8001,yourhost1.com:8002,yourhost2.com:7030
```

Click **OK**.

Finally, if OSM is not deployed in a clustered WebLogic Server environment, enter the name of the WebLogic server where service order management is running, and click **OK**.

j. Click **OK**. The system configures the rest of the WebLogic resources. This may take a few minutes.

5. When the installer is finished, shut down any affected WebLogic domains and restart them.
6. Exit and restart Eclipse.

> **Note**
>
> If you have made a mistake setting the Design Studio preferences and it causes this procedure to fail, this will be displayed in the Console view in Design Studio. First, correct the preferences using the instructions in "Setting Design Studio Preferences." Next, go to the **Properties** tab of the Edit Configuration dialog box, select **Use global properties as specified in the Ant runtime preferences** to update the values, and then deselect **Use global properties as specified in the Ant runtime preferences** again. Then, select **Clean** from the **Project** menu and clean and build the OracleComms_OSM_O2A_Install project. Exit and restart Design Studio, and then begin the procedure for configuring the WebLogic Server resources again.

Deploying the Updated Order-to-Activate Cartridges

The following section provides the procedure for deploying the updated Order-to-Activate cartridges. It must be performed for all cartridge updates.

Before deploying the Order-to-Activate cartridges:

- Ensure that all of the cartridges have been built successfully. These steps are included in the individual procedures for updating the Order-to-Activate cartridges.
- Configure the WebLogic Server resources.

To deploy the updated Order-to-Activate cartridges:

1. Verify that Design Studio is running.
2. Create a new Studio Environment project and a new Studio environment.

> **Note**
>
> See the Design Studio Help for details on creating a Studio Environment project and a Studio environment.
>
> If you are installing central order management and service order management on different OSM servers, you need two environment entities: one pointing to the central order management cluster or standalone server and the other pointing to the service order management cluster or standalone server.

3. Open the Environment perspective.
4. Select the Design Studio environment in which the cartridges are to be deployed.

The Cartridge Management pane is displayed, displaying a list of available cartridges.

5. Using the appropriate table below, select all of the cartridges you wish to deploy and click **Deploy**.

   Table 12-3 describes which cartridges to deploy if you are using the Order-to-Activate 2.1.0.x or 2.1.2.x cartridges:

Table 12-3 Cartridge to Deploy for Order-to-Activate 2.1.0.x or 2.1.2.x

Using Calculate Service Order Option?	Current Workspace Is for:	Topology	Cartridge to Deploy from Workspace
Yes	COM only	All	OracleComms_OSM_O2A_COM_CSO_Solution
Yes	SOM only	All	OracleComms_OSM_O2A_SOM_CSO_Solution
Yes	COM and SOM	All	OracleComms_OSM_O2A_COMSOM_CSO_Solution
No	COM only	Simple	OracleComms_OSM_O2A_COM_Simple_NP_Soln
No	COM only	Typical or Complex	OracleComms_OSM_O2A_COM_Typical_NP_Soln
No	SOM only	All	OracleComms_OSM_O2A_SOM_NP_Soln
No	COM and SOM	Simple	OracleComms_OSM_O2A_COMSOM_Simple_NP_Soln
No	COM and SOM	Typical or Complex	OracleComms_OSM_O2A_COMSOM_Typical_NP_Soln

Table 12-4 describes which cartridges to deploy if you are using the Order-to-Activate 2.0.1.x or 7.2 cartridges:

Table 12-4 Cartridges to Deploy for Order-to-Activate 2.0.1.x or 7.2

Topology	Configuration	Cartridges to deploy
Simple	Central order management and service order management are on **the same** OSM instance	OracleComms_OSM_O2A_COMSOM_SimpleSolution
Simple	Central order management and service order management are on **different** OSM instances	OracleComms_OSM_O2A_COM_SimpleSolution (Deploy to central order management environment.) OracleComms_OSM_O2A_SOM_Solution (Deploy to service order management environment.)
Typical	Central order management and service order management are on **the same** OSM instance	OracleComms_OSM_O2A_COMSOM_TypicalSolution
Typical	Central order management and service order management are on **different** OSM instances	OracleComms_OSM_O2A_COM_TypicalSolution (Deploy to central order management environment.) OracleComms_OSM_O2A_SOM_Solution (Deploy to service order management environment.)

Updating Order-to-Activate Cartridges Manually

This section provides instructions for updating Order-to-Activate cartridges manually.

Updating the Order-to-Activate 2.1.2.x Cartridges

To update Order-to-Activate 2.1.2.x cartridges:

1. Import the Order-to-Activate 2.1.2.x cartridges into Design Studio.
2. Go to the Studio Projects view.
3. For each project, select the **Target Version** as **7.4.0** or the version closest to the actual OSM version but not newer than it.

   If the project is sealed, click **Unseal** to unseal it and then change the Target Version.
4. Restart the workspace.

   The workspace shows up and the studio upgrade process is triggered.
5. Click **Finish** to upgrade the projects.
6. Restart Design Studio.
7. Using ant view, run “config_all" target and make sure that the resources are created in the WebLogic server and the necessary **MetaData & Model** variable is set.
8. Deploy emulators, if you are using them and restart the Weblogic server.
9. Clean and build the cartridges.
10. Deploy the cartridges and submit sample orders.

Updating the Order-to-Activate 2.1.0.2.x and 2.1.0.1.x Cartridges

To update Order-to-Activate 2.1.0.2.x and 2.1.0.1.x cartridges:

1. Import the Order-to-Activate 2.1.0.2.x and 2.1.0.1.x cartridges into Design Studio.

   Design Studio displays error markers for the missing JAR files.
2. For each cartridge against which an error marker is shown in the build path, modify the JAR file name to the new name as specified in Table 12-5.

Table 12-5 Old and New JAR Filenames

Old JAR Filename	New JAR Filename
ORACLE_MIDDLEWARE_HOME\modules\javax.jms_1.1.1.jar	*ORACLE_MIDDLEWARE_HOME*\oracle_common\modules\javax.jms.javax.jms-api.jar
ORACLE_MIDDLEWARE_HOME\oracle_common\modules\oracle.jps_11.1.1\jps-api.jar	*ORACLE_MIDDLEWARE_HOME*\oracle_common\modules\oracle.jps\jps-api.jar
ORACLE_MIDDLEWARE_HOME\wlserver_10.3\server\lib\weblogic.jar	*ORACLE_MIDDLEWARE_HOME*\wlserver\server\lib\weblogic.jar
ORACLE_MIDDLEWARE_HOME\wlserver_10.3\server\lib\wlclient.jar	*ORACLE_MIDDLEWARE_HOME*\wlserver\server\lib\wlclient.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\commons-collections-3.2.1.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-collections4.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\ commons-lang3-3.1.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-lang3.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\commons-logging-1.1.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-logging.jar

Table 12-5 (Cont.) Old and New JAR Filenames

Old JAR Filename	New JAR Filename
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\saxon-ee-java.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9-xpath.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\saxon-ee-java-osm-license.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9-dom.jar	Remove this JAR file.

3. Go to the Studio Projects view.
4. For each project, select the **Target Version** as **7.4.0** or the version closest to the actual OSM version but not newer than it.

   If the project is sealed, click **Unseal** to unseal it and then change the Target Version.
5. Restart the workspace.

   The workspace shows up and the studio upgrade process is triggered.
6. Click **Finish** to upgrade the projects.
7. Clean and build the cartridges.

   Design Studio displays the following errors:

   - Automation Build Error: Cartridge *OracleComms_OSM_O2A_COM_Base* has an automation build file (automationBuild.xml). Automation build files are not supported for target version 7.4.0. Right click on problem marker and select Quick Fix.
   - Automation Build Error: The automationMap.xsd file in cartridge *OracleComms_OSM_O2A_COMSOM_CSO_Solution* is incorrect for this Design Studio release. Right click on the problem marker, select Quick Fix, and apply the Automation Quick Fix to correct.
   - Composite Cartridge Model Error: In the Order Item specifications [COM_SalesOrderLine and SOM_ProvisionOrderLine], Dynamic Parameter Property is either not set OR set to a different value. This property must be same for all Order Item specifications in the cartridges.

     This error marker appears for non-CSO cartridges in Design Studio 7.4.1. This is due to the validation introduced in Design Studio 7.4.1. To resolve this issue, perform the following steps:

     – Open **OracleComms_OSM_O2A_SOM_Base\Data Schemas\OracleComms_OSM_O2A_SOM_Base** and do the following:

       1. Right- click OrderItem and select **Add Child Structure**.

       2. In the **Child Structure** dialog box, for Type, click **Select**.

       3. In the **Select Base Type** dialog box, clear the Filter Project Dependencies check box and select **dynamicParams** in the list.

       5. Click **Finish**. This adds dynamicParams under OrderItem.

     – Open **OracleComms_OSM_O2A_SOM_Base\Order Item Specifications\ORDER_ITEM\SOM_ProvisionOrderLine** and do the following:

       1. In Order Item Properties, add **dynamicParams**.

       2. In Property Reference, select **dynamicParams** for Dynamic Parameter Property.

3. In Order Template, expand ControlData and right-click OrderItem.
4. Select **Select From Dictionary** and select **OrderItem\dynamicParams**.
5. Click **Finish**. This adds **dynamicParams** under ControlData\OrderItem.

– Open **OracleComms_OSM_O2A_SOM_Base\Manual Tasks\TASK_CREATION\SOM_ProvisionOrderFulfillment_CreationTask** and do the following:

1. Right click **ControlData\OrderItem** and select **Select from Order Template**.
2. Select **ControlData\OrderItem\dynamicParams** and click **Finish**. This adds **dynamicParams** under ControlData\OrderItem\dynamicParams.

8. Restart the workspace.
9. Update the **OracleComms_OSM_O2A_Configuration\etc\Manifest.mf** file to have the right set of JAR files. (As per the above JAR file name changes table).
10. Copy the following files from Cartridge Migration Project (**OracleComms_OSM_O2A_CartridgeMigration***<2.1.0.2 or 2.1.0.1>* **OracleComms_OSM_O2A_Configuration**) to **OracleComms_OSM_O2A_Configuration** in your workspace.
    - OSM.O2A.CartridgeConfigUtility.xml
    - OSM.O2A.EmulatorsBuild.xml
    - OSM.O2A.Env.xml

    > **Note**
    >
    > If you have modified these files, ensure that you compare the files available in **OracleComms_OSM_O2A_CartridgeMigration***<2.1.0.2 or 2.1.0.1>* with your files and include the changes added in the migration set into the cartridges.

11. Click **Save All** and restart Design Studio.
12. Using ant view, run “config_all" target and make sure that the resources are created in the WebLogic server and the necessary **MetaData & Model** variable is set.
13. Deploy emulators, if you are using them and restart the Weblogic server.
14. Clean and build the cartridges.
15. Deploy the cartridges and submit sample orders.

Updating Order-to-Activate 2.0.1.x Cartridges

To update Order-to-Activate 2.0.1.x cartridges:

1. Import the Order-to-Activate 2.0.1.x cartridges into Design Studio.

   Design Studio displays error markers for the missing JAR files.
2. For each cartridge against which an error marker is shown, in the build path, modify the JAR file name to the new name as specified in Table 12-6.

Table 12-6 Old and New JAR File Names

Old JAR Filename	New JAR Filename
ORACLE_MIDDLEWARE_HOME\modules\javax.jms_1.1.1.jar	*ORACLE_MIDDLEWARE_HOME*\oracle_common\modules\javax.jms.javax.jms-api.jar
ORACLE_MIDDLEWARE_HOME\oracle_common\modules\oracle.jps_11.1.1\jps-api.jar	*ORACLE_MIDDLEWARE_HOME*\oracle_common\modules\oracle.jps\jps-api.jar
ORACLE_MIDDLEWARE_HOME\modules\javax.ejb_3.0.1.jar	*ORACLE_MIDDLEWARE_HOME*\oracle_common\modules\javax.ejb.javax.ejb-api.jar
ORACLE_MIDDLEWARE_HOME\wlserver_10.3\server\lib\weblogic.jar	*ORACLE_MIDDLEWARE_HOME*\wlserver\server\lib\weblogic.jar
ORACLE_MIDDLEWARE_HOME\wlserver_10.3\server\lib\wlclient.jar	*ORACLE_MIDDLEWARE_HOME*\wlserver\server\lib\wlclient.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\commons-collections-3.2.1.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-collections4.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\commons-lang-2.6.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-lang3.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\commons-logging-1.1.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-logging.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\saxon-ee-java.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9-xpath.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\saxon-ee-java-osm-license.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9-dom.jar	Remove this JAR file.

3. Go to the Studio Projects view.
4. For each project, select the **Target Version** as **7.4.0** or the version closest to the actual OSM version but not newer than it.

   If the project is sealed, click **Unseal** to unseal it and then change the Target Version.
5. Restart the workspace.

   The workspace shows up and the studio upgrade process is triggered.
6. Click **Finish** to upgrade the projects.
7. Clean and build the cartridges.

   Design Studio displays three types of error markers.

   - Automation Build Error: Cartridge **OracleComms_OSM_O2A_SOM_Base** has an automation build file (automationBuild.xml). Automation build files are not supported for target version 7.4.0. Right click on problem marker for the Quick Fix.
   - Base Task Control Data: Base Task FulfillBillingBaseTask must contain the node / ControlData/Functions/FulfillBillingFunction/calculatedStartDate.
   - Base Task Control Data: Base Task FulfillBillingBaseTask must contain the node / ControlData/Functions/FulfillBillingFunction/duration.
8. For each error marker, right click and select **Quick Fix** and apply the default options.

   Design Studio applies the necessary changes in the cartridges to add **calculatedStartDate** and **duration**.
9. Restart the workspace.

10. Copy the **osm-o2a-utility.jar** file from Cartridge Migration Project (**OracleComms_OSM_O2A_CartridgesMigration\2.0.1.3\OracleComms_OSM_O2A_CommonUtility\resources**) to the current workspace (**OracleComms_OSM_O2A_CommonUtility\resources**).
11. Copy the **o2a-saxonee9-config_2.0.1.jar** file from Cartridge Migration Project (**OracleComms_OSM_O2A_CartridgesMigration\2.0.1.3\OracleComms_OSM_O2A_Configuration\o2alib**) to the current workspace (**OracleComms_OSM_O2A_Configuration\o2alib**).
12. Add the **o2alib\o2a-saxonee9-config_2.0.1.jar** file to the build class path of the OracleComms_OSM_O2A_Configuration cartridge in Eclipse/Design Studio.
13. Update the **OracleComms_OSM_O2A_Configuration\etc\Manifest.mf** file to have the right set of JAR files (as per the above JAR file name changes table).
14. Copy the following files from Cartridge Migration Project (**OracleComms_OSM_O2A_CartridgeMigration\2.0.1.3\OracleComms_OSM_O2A_Configuration**) to **OracleComms_OSM_O2A_Configuration** in your workspace.
    - src\oracle\communications\ordermanagement\util\transform\XQueryTransformer.java
    - src\oracle\communications\ordermanagement\aiaemulator\AIAMultipleResponseEmulatorBean.java
    - OSM.O2A.CartridgeConfigUtility.xml
    - OSM.O2A.EmulatorsBuild.xml
    - OSM.O2A.Env.xml

    > **Note**
    >
    > If you have modified these files, ensure that you compare the files available in **OracleComms_OSM_O2A_CartridgeMigration\2.0.1.3** with your files and include the changes added in the migration set into the cartridges.

15. Perform the following changes in the **OracleComms_OSM_O2A_COM_Base** cartridge. These changes are done and available in the OracleComms_OSM_O2A_CartridgeMigration\2.0.1.3 cartridges.
    a. Copy the model task **OracleComms_OSM_O2A_CartridgeMigration\2.0.1.3\OracleComms_OSM_O2A_COM_Base\model\TASK_QUERY\COM_OrphanOrderItemInitialFulfillmentStateView.manualTask_migrated** to your workspace. OSM uses the manual task **COM_OrphanOrderItemInitialFulfillmentStateView** as a Query Task in the COM order.
    b. Right-click the folder and refresh.
    c. Open the **COM_SalesOrderFulfillment** order.
    d. Select the Permissions tab and then select the Query Tasks tab.
    e. Click **Add** and select **COM_OrphanOrderItemInitialFulfillmentStateView**.
    f. Click **Save**.

    The following extract shows the changes in the **OracleComms_OSM_O2A_COM_SalesOrderFulfillment** cartridge. The changes are available in the **resources\FulfillmentState\FulfillmentStateModule.xquery** file.

```
[...]
declare
variable $fulfillmentstatemodule:COM_ORDER_ORPHAN_ORDERITEM_INITIAL_FULFILLMENTSTATE_
VIEW := "COM_OrphanOrderItemInitialFulfillmentStateView"; declare
variable $fulfillmentstatemodule:CANCEL := "CANCEL";
[...]
(:
 : Function to return the component fulfillment state path for order item
 : By default, return "" to disable.
 : Note: Solution provide version should be a properties under
/ControlData/OrderItem/___the_property_name___
:)
declare function
fulfillmentstatemodule:getOrderItemComponentFulfillmentStatePath( $orderMnemonic as
xs:string) as xs:string
{
    ""
};

(:
 : Function to return the component fulfillment state description path for order
item
 : By default, return "" to disable.
 : Note: Solution provide version should be a properties under /ControlData/
OrderItem/___the_property_name___
:)
declare function
fulfillmentstatemodule:getOrderItemComponentFulfillmentStateDescriptionPath( $orderMn
emonic as xs:string) as xs:string
{
    ""
};

(:
 : Function to return the component Id for order item
 : By default, return "" to disable.
 : Note: Solution provide version should be a properties under /ControlData/
OrderItem/___the_property_name___
 :)
declare function fulfillmentstatemodule:getOrderItemComponentIdPath( $orderMnemonic
as xs:string) as xs:string
{
    ""
};
[...]
(:
 : Function to return the name of the View/QueryTask to be used when determining
initial fulfillment state for orphan order items. The returned view name will be
used to fetch the order data, which is then passed to the XQuery extension function
getOrphanOrderItemInitialFulfillmentState(). If there is no need to calculate
initial orphan order item fulfillment states then an empty string can be returned.
:)
declare function
fulfillmentstatemodule:getOrphanOrderItemInitialFulfillmentStateView( $orderMnemonic
as xs:string) as xs:string
{
    if($orderMnemonic = $fulfillmentstatemodule:COM_ORDER)
then $fulfillmentstatemodule:COM_ORDER_ORPHAN_ORDERITEM_INITIAL_FULFILLMENTSTATE_VIEW
 else ""
};

(:
```

```
 : Function to detect is the current operation of the order is a Cancelation
Operation
 :)
declare function fulfillmentstatemodule:isCancelRevision( $taskData as element()) as
xs:boolean
{
    let $osmOperationType := fn:normalize-space($taskData/oms:_root/
oms:CustomerHeaders/oms:OsmOperationType/text())
return
    if ($osmOperationType=$fulfillmentstatemodule:CANCEL)
    then fn:true()
    else fulfillmentstatemodule:isCanceledByAdmin($taskData)
};

(:
 : Function to detect if the cancel is triggered by OSM Admin from Web GUI
 :)
declare function fulfillmentstatemodule:isCanceledByAdmin( $taskData as element())
as xs:boolean
{
    let $orderState := fn:normalize-space($taskData/oms:OrderState/text())
    return
        if ($orderState = ("open.running.compensating.cancelling",
"open.not_running.cancelled"))
        then fn:true()
        else fn:false()
};

(:
 : Function to return the namespace and name of the initial FulfillmenState. The
format for the result is as a clark name which is {namespace}localName.
 : e.g. {demo.com}completed
 : If there is no initial fulfillment state, then an empty string can be returned.
 :)
declare function
fulfillmentstatemodule:getOrphanOrderItemInitialFulfillmentState( $order as document-
node()) as xs:string
{
    let $orderData := $order/oms:GetOrder.Response
    let $orderMnemonic := fn:normalize-space($orderData/oms:OrderSource/text())
    return
        if($orderMnemonic = $fulfillmentstatemodule:COM_ORDER)
        then
        (
            let $isCancel := fulfillmentstatemodule:isCancelRevision($orderData)
            let $orphanOrderItemFulfillmentState :=
                if ($isCancel = fn:true())
            then
fn:concat("{",$fulfillmentstatemodule:COM_ORDER_INITIAL_FULFILLMENTSTATE_NAMESPACE,"}
", $o2acomfulfillmentstate:CANCELLED_STATE)
            else
fn:concat("{",$fulfillmentstatemodule:COM_ORDER_INITIAL_FULFILLMENTSTATE_NAMESPACE,"}
", $o2acomfulfillmentstate:COMPLETE_STATE)
        return
            $orphanOrderItemFulfillmentState
        )
        else ""
};
[...]
    $orderData as element(),
    $orderItemIndexes as xs:integer*) as element()?
[...]
```

16. In the Saxon API call, make the following changes:

    - Changed method in the file: **OracleComms_OSM_O2A_COM_Base\ resources\ OrderStateHandler\ OrderFailedStateHandler.xquery**

```
declare function local:getAttachedSalesOrderEBM() as element()?
{
    let $names := context:getAllAttachmentFileNames($context)
    return
        if (fn:exists($names))
        then
        (
            let $name := $names[1]
            return
                if (fn:exists($name))
                then saxon:parse(context:getAttachmentAsString($context,
xs:string($name)))/salesord:ProcessSalesOrderFulfillmentEBM
                else ()
        )
        else ()
};
```

    - Changed method in the file: **OracleComms_OSM_O2A_SOM_Base\ resources\ OrderStateHandler\ LFCheckCreationOrderFailure.xqy**

```
declare function local:getAttachedProvOrderEBM() as element()*
{
    let $names := context:getAllAttachmentFileNames($context)
    return
        if (fn:exists($names)) then
        (
            let $name := $names[1]
            return
                if (fn:exists($name)) then
                (
                    saxon:parse(context:getAttachmentAsString($context,
xs:string($name)))/provord:ProcessProvisioningOrderEBM
                )
              else ()
        )
        else ()
};
```

17. Click **Save All** and restart Design Studio.
18. Using ant view, run “config_all" target and make sure that the resources are created in the WebLogic server and the necessary **MetaData & Model** variable is set.
19. Deploy emulators, if you are using them and restart the Weblogic server.
20. Clean and build the cartridges.
21. Deploy the cartridges and submit sample orders.

Updating the Order-to-Activate 7.2.0.x Cartridges

To update Order-to-Activate 7.2.0.x cartridges:

1. Import the Order-to-Activate 7.2.0.x cartridges into Design Studio.

   Design Studio displays error markers for the missing JAR files.
2. For each cartridge against which an error marker is shown, modify the JAR file names in the build paths as per the following table:

Table 12-7 Old and New JAR Filenames

Old JAR Filename	New JAR Filename
ORACLE_MIDDLEWARE_HOME\modules\javax.jms_1.1.1.jar	*ORACLE_MIDDLEWARE_HOME*\oracle_common\modules\javax.jms.javax.jms-api.jar
ORACLE_MIDDLEWARE_HOME\oracle_common\modules\oracle.jps_11.1.1\jps-api.jar	*ORACLE_MIDDLEWARE_HOME*\oracle_common\modules\oracle.jps\jps-api.jar
ORACLE_MIDDLEWARE_HOME/modules/javax.ejb_3.0.1.jar	*ORACLE_MIDDLEWARE_HOME*/oracle_common/modules/javax.ejb.javax.ejb-api.jar
ORACLE_MIDDLEWARE_HOME\wlserver_10.3\server\lib\weblogic.jar	*ORACLE_MIDDLEWARE_HOME*\wlserver\server\lib\weblogic.jar
ORACLE_MIDDLEWARE_HOME\wlserver_10.3\server\lib\wlclient.jar	*ORACLE_MIDDLEWARE_HOME*\wlserver\server\lib\wlclient.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\commons-collections-3.2.1.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-collections4.jar
OSM_SDK_HOME/Automation/automationdeploy_bin/commons-lang-2.6.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-lang3.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\commons-logging-1.1.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\commons-logging.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\saxon-ee-java.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9-xpath.jar	*OSM_SDK_HOME*\Automation\automationdeploy_bin\saxon-ee-java-osm-license.jar
OSM_SDK_HOME\Automation\automationdeploy_bin\saxon9-dom.jar	Remove this JAR file.

3. Go to the Studio Projects view.
4. For each project, select the **Target Version** as **7.4.0** or the version closest to the actual OSM version but not newer than it.

   If the project is sealed, click **Unseal** to unseal it and then change the Target Version.
5. Restart the workspace.

   The workspace shows up and the studio upgrade process is triggered.
6. Click **Finish** to upgrade the projects.
7. Clean and build the cartridges.

   Design Studio displays three types of error markers.

   - Automation Build Error: Cartridge *OracleComms_OSM_O2A_SOM_Base* has an automation build file (automationBuild.xml). Automation build files are not supported for target version 7.4.0. Right click on problem marker for the Quick Fix.
   - Creation Task Control Data: Creation Task COM_SalesOrderFulfillment_CreationTask must contain the node ControlData/Functions/FulfillBillingFunction/calculatedStartDate.
   - Creation Task Control Data: Creation Task COM_SalesOrderFulfillment_CreationTask must contain the node ControlData/Functions/FulfillBillingFunction/duration.
8. For each error marker, right click and select **Quick Fix** and apply the default options.
9. Restart the workspace.

10. Update the **OracleComms_OSM_O2A_Configuration\etc\Manifest.mf** file to have the right set of JAR files. (As per the above JAR file name changes table).
11. Copy the files listed in Table 12-8 from Cartridge Migration Project (**OracleComms_OSM_O2A_CartridgeMigration\7.2.0**) to the respective cartridges in your workspace.

Table 12-8 Cartridge Migration Project Files

Cartridge Name	File Name
OracleComms_OSM_O2A_CommonUtility	• resources\FulfillmentOrderLifecycleManagementModule.xquery • resources\OrderLifecycleModule.xquery • resources\osm-o2a-utility.jar
OracleComms_OSM_O2A_COM_SalesOrderFulfillment	• resouces\FulfillmentState\FulfillmentStateModule.xquery • resources/ComponentInteraction/AIAEBMRequest_do.xqy • resources/ComponentInteraction/AIAEBMRequest_redo.xqy • resources/ComponentInteraction/AIAEBMRequest_undo.xqy • resources/ComponentInteraction/AIAEBMResponse.xqy • resources/ComponentInteraction/InitiateWaitforProvisioningResponse.xqy • resources/ComponentInteraction/OrderLifecycleManagementModule.xquery • resources/ComponentInteraction/SIEntryPoint.xqy • resources/ComponentInteraction/SIExitPoint.xqy • resources/ComponentInteraction/SIMilestone_doredo.xqy • resources/ComponentInteraction/UpdateSalesOrderStatusFunctions.xqy
OracleComms_OSM_O2A_Configuration	• OSM.O2A.CartridgeConfigUtility.xml • OSM.O2A.EmulatorsBuild.xml • src/oracle/communications/ordermanagement/util/transform/XQueryTransformer.java • xslt/xmlie-config.xsl
OracleComms_OSM_O2A_RecognitionFallout	resources/CreateErrorFault.xqy
OracleComms_OSM_O2A_SOM_Base	resources/OrderStateHandler/LFCheckCreationOrderFailure.xqy

> **Note**
>
> If you have modified these files, ensure that you compare the files available in **OracleComms_OSM_O2A_CartridgeMigration\7.2.0** with your files and include the changes added in the migration set into the cartridges.

12. Click **Save All** and restart Design Studio.
13. Using ant view, run “config_all" target and make sure that the resources are created in the WebLogic server and the necessary **MetaData & Model** variable is set.

14. Deploy emulators, if you are using them and restart the Weblogic server.
15. Clean and build the cartridges.
16. Deploy the cartridges and submit sample orders.

13

Uninstalling OSM

This chapter provides information about uninstalling Oracle Communications Order and Service Management (OSM).

The OSM uninstallation process removes the OSM program files. It leaves the database schemas, WebLogic configuration, and the deployed OSM application unchanged.

To completely remove OSM, you must perform the following steps:

1. Back up all customized configuration files.
2. Remove the OSM application from the WebLogic server. See "OSM Uninstall: Additional Tasks" for minimal instructions, and see the Oracle WebLogic Server documentation for more information.
3. Remove the OSM database schemas. See Oracle Database documentation for more information.
4. Remove the directory where OSM is installed.

Uninstalling OSM Components

To uninstall OSM, run the following commands in command shell:

Using RPM

```
sudo rpm -e package_name
```

Where *package_name* is the OSM installer name with the version number and build number.

This command deletes all files installed by the RPM package.

Using DNF

```
sudo dnf remove --installroot=/path/to/installationdir
installed_package_name --nobest --skip-broken --setopt protected_packages=
```

Where *installed_package_name* is the OSM installer name with the sample OS Architecture extension (for example, `osm-installer.x86_64`).

This command deletes all files installed by the DNF package. After successful uninstallation using DNF, the entire **/path/to/installationdir** directory can be deleted manually, including any residual content.

To know the `installed_package_name`, run the following command:

```
sudo dnf list --installroot=/path/to/installationdir
```

OSM Environment Files
The configuration directory, **$OSM_CFG_HOME**, contains a subdirectory named after the OSM environment. This subdirectory's contents represent the installer's state information for this

environment. These files are not affected by the above steps and can be retained for reference or to recreate the environment. If this environment is not required, you can delete these files manually.

OSM Uninstall: Additional Tasks

In addition to running the OSM software uninstall procedure, you must perform additional tasks to completely remove the OSM software and related software elements.

You should perform an environment-specific analysis on each OSM system to create a customized list of uninstallation tasks. For example, you may want to remove customizations that you added after OSM was installed. These customizations might be part of a clustered environment or a multiproduct installation. You may also need to remove additional dependent WebLogic resources, such as messaging bridges, JMS Queues, automation plug-ins, and user accounts.

The specific steps for uninstalling all related elements will vary based on your particular environment.

The following procedure specifies the additional uninstallation tasks for a minimal OSM installation in a single server environment:

1. Close the OSM applications running in the system, including:
   - Oracle Communications Design Studio
   - XML Import/Export application
   - XML API agents
2. Stop the OSM WebLogic applications.
3. Log in to the WebLogic Remote console. Select **Edit Tree**
4. Click **Deployments**, and stop the OSM application **oms**, the cartridge management web services application **cartridge_management_ws**, and the shared Java EE library **comms-platform-webapp**.

   > **Note**
   >
   > Do not stop **cartridge_management_ws** and **comms-platform-webapp** in a suite or multiproduct environment because other applications may be dependent on them.

5. Click the OSM application **oms**, then select the **Configuration** tab, then the **Workload** tab, and delete all of the application-scoped work managers.
6. Navigate to **Services**, then **Messaging**, then **JMS Modules**, and delete **oms_jms_module**.
7. Navigate to **Services**, then **Messaging**, then **JMS Servers**, and delete **oms_jms_server**.
8. Navigate to **Services**, then **Persistent Stores**, and delete **oms_jms_store**, if you have configured it.
9. Navigate to **Services**, then **JDBC**, then **Data Sources**, and delete **oms_pool**.
10. Navigate to **Services**, then **File T3**, and delete **oms_remote_file_system**.
11. Navigate to **Security Realms**, then **myrealm**, then **Users and Groups**, then **Users** and delete the following OSM users:

- **oms-automation**
- **oms-internal**
- **deployAdmin**
- **admin**

> **Note**
>
> **admin** is the default OSM Administrator user and **deployAdmin** is the default Design Studio administrator user. If you specified different names during the OSM installation, remove the custom users.

12. Navigate to **Security Realms**, then **myrealm**, then **Users and Groups**, then **Groups**, and delete the following OSM user groups if present:
    - **Cartridge_Management_WebService**

    > **Note**
    >
    > Do not delete **Cartridge_Management_WebService** in a suite or multiproduct environment because other applications may be dependent on it.

    - **OMS_cache_manager**
    - **OMS_client**
    - **OMS_log_manager**
    - **OMS_user_assigner**
    - **OMS_workgroup_manager**
    - **OMS_ws_api**
    - **OMS_ws_diag**
    - **OMS_xml_api**
    - **OSM_automation**
    - **osmRestApiGroup**
    - **osmEntityClientGroup**
13. Click **Deployments**, and delete the OSM application **oms**, the cartridge management web services application **cartridge_management_ws**, and the shared Java EE library **comms-platform-webapp**.

    > **Note**
    >
    > Do not delete **cartridge_management_ws** and **comms-platform-webapp** in a suite or multiproduct environment because other applications may be dependent on them.

14. Close the WebLogic server.
15. Navigate to the **Domain** directory, and delete the **Attachments** directory.

A

Production Readiness Checklist

This appendix provides a checklist of tasks that must be accomplished before taking Oracle Communications Order and Service Management (OSM) into production.

About Using the Production Checklist

Use the OSM checklist to verify that operational best-practices and key procedures exist before to going into production or performing a major release upgrade. These operational best-practices include success factors such as performance, configuration, purging, and database statistics management.

Ensure that implementation project teams review this material well ahead of going into production to ensure that project plans incorporate the required activities.

Ensure that implementation project teams provide an explicit response to this checklist to Oracle. The production checklist can help Oracle clarify key success factors and risks so that Oracle can provide faster emergency responses in case of problems.

The implementation project should share additional information (for example, performance test results) with Oracle to provide better overall context on the deployment. Oracle maintains all such information confidentially according to the Oracle Information Protection Policy. Shared information can be deleted upon request.

Provide the checklist response and any additional information by opening an OSM service request with a title such as “Production Checklist" at least one month prior to going into production. Oracle may request clarifications through the service request, which can then be closed.

Checking for a Current OSM Patch Before Going into Production

Going into production using a recent OSM patch ensures that the OSM solution is not susceptible to known issues encountered by other customers, which may or may not manifest themselves in pre-production tests. It also facilitates resolution of any new issues that may be encountered, which are normally resolved on recent patches.

For OSM releases in Premier Support with active customers, Oracle continue to release patch sets (for example, 7.3.0.1.0, or 7.3.0.2.0) at a regular interval. The patch set interval depends on the level of customer activity for a given release. For example, the patch set interval may be every six months. The majority of bugs are fixed in patch sets and often include important operational enhancements (for example, improved purging performance).

In addition, Oracle releases interim patches (for example, 7.3.0.1.2) to address specific problems that are blocking specific customers. These interim patches are released on a code branch for the given patch set. All interim patches are cumulative (a superset) of prior interim patches for that branch. Changes released in an interim patch are forwarded-ported into the next upcoming patch set.

Oracle normally releases interim patches for the last two patch sets of a given release.

The implementation project teams should plan to go into production or upgrade on a recent patch of the latest patch set (preferred) or the prior patch set (also acceptable). For example, if

7.3.0.1.0 was the last patch set issued, going into production on 7.3.0.1.5 (a recent patch of that patch set) would be the preferred option. Alternatively, it would also be acceptable to go into production on 7.3.0.0.14, a recent patch of 7.3.0.0.0 (the prior patch set).

In general, if the implementation project takes more than six months, Oracle recommends that the implementation project teams plan to upgrade to a recent patch on the same release prior to going into production. Oracle can provide advanced notice of planned patch sets to assist in this planning. For example, if a project started three months ago using OSM 7.3.0.0.9 (on the prior patch set) and the implementation project teams plan to go into production in the next three months, Oracle would recommend that the teams check with Oracle when a 7.3.0.1.0 patch set may be released. The implementation project teams should plan to upgrade prior to going into production, avoiding going into production on an older patch set.

Checklist:

- The production system launch or upgrade should take place on a recent interim patch for the latest patch set (preferred) or the previous patch set.
- If implementation project teams have a long-running project (more than six months), plan to upgrade to a patch of the most recent patch set prior to going into production or upgrading.

Useful information to share: What is the five digit OSM patch level targeted for going into production or upgrading?

Checking for Deployment Architecture

Understanding the OSM deployment architecture is important in troubleshooting configuration-related production issue. Diagrams depicting deployment architecture information are helpful.

Checklist:

- Optimize storage I/O to sustain both order processing and order (row-based) purging. For more information, see "RAID Recommendations for the Database".
- Consider using fast solid state drives (SSD) disks for critical database files (for example, REDO logs). For more information, see "RAID Recommendations for the Database".
- Validate that shared WebLogic Server persistent store storage I/O is enough for the order processing throughput. For high-volume deployments fast SSD are recommended. For more information, see "Network Latency and NFS Configuration for WebLogic Server Shared Storage".
- Application logs (including WebLogic Server and OSM) are written to a local file-system.

Useful information to share:

- At a high-level, what are the external systems that OSM interacts with?
- How many application servers are deployed? What are their hardware specifications?
- Is Oracle RAC used and if so what Oracle RAC configuration is used (for example, Active/Active)?
- How many database servers are deployed and what are their hardware specifications?
- Are virtual machines used, if so which product and version?
- Are multiple instances of OSM (for example, COM, SOM) deployed?
- What are the version numbers for key other software (Oracle Application Integration Architecture (Oracle AIA), OSM Order-to-Activate cartridges, WebLogic Server, operating system, database, virtual machine)?

- What RAID storage configuration is used (for example RAID 10, RAID 5)?
- What type of storage is used for the shared WebLogic Server Persistent Store?
- What security provider configuration (for example, internal or external LDAP directory) does OSM use?

Checking the OSM Production System Configuration

Optimal performance and system behavior under load or faults depend heavily on proper configuration of OSM, the database, and WebLogic Server. The compliance tool identifies configuration problems that may cause production issues. Run the compliance tool before doing performance tests to avoid costly delays. For more information about running the compliance tool, see "Verifying the OSM Installation".

Checklist:

- The compliance tool was run on the OSM production system.
- All configuration compliance issues of severity “Error" and “Warning" have been corrected or reviewed and determined to be acceptable.
- The size of log files and their rollover is correctly configured for app server and database components according to the amount of disk space available.
- Do a manual review of all configuration files to identify any typos or inconsistent configurations (for example, between managed server startup parameters, and so on).
- Ensure that all recommended patches have been installed. For more information, see "Software Requirements".

Useful information to share: Rerun the compliance tool when all corrections have been made and share the report with Oracle. Add an explanation for any “Error" or “Warning" that are ignored.

Checking for Performance Expectations

Ensure that you validate that the OSM solution performs at the full expected target order volume prior to going into production. For more information about running performance tests on the OSM solution, see "OSM Pre-Production Testing and Tuning".

Gathering this information ahead of time may save precious investigation time to resolve a production issue.

Checklist:

- Performance tests have confirmed that the OSM production system can sustain the full expected maximum hourly order volume rate. These tests should observe the following:
  - Ensure that representative mix of order types and size is used in the test workload.
  - Ensure that the average size of the order (line item count for orchestration orders) is aligned with expected production order workload.
  - Ensure that the test order mix includes enough revision orders and larger orders, if applicable.
  - Ensure that the production system has enough memory to handle the expected number of cartridge versions deployed.
  - Ensure that for long-running orders, enough in-progress orders are already loaded in the system to simulate the maximum workload of the system.

  - Validate that enough capacity remains to sustain peak volume if any one machine of the production system were to fail, or to deal with unexpected order volume peaks (for example, +20%).
- Performance tests have confirmed that the production system can process the maximum expected order size.
- Performance tests have confirmed that the OSM production system can process any expected large bulk operations.
- Longevity performance tests (for example, 24 hours) have been performed without errors on the OSM production system. Longevity performance tests must validate the following:
  - The workload is distributed uniformly across the cluster.
  - JMS messages do not accumulate.
  - Java Garbage collection is operating properly. Oracle recommends that you perform a regular analysis of garbage collection logs (for example, with GCViewer) to detect changes in your memory usage patterns. In particular, frequently check the live data size (LDS), which is the amount of memory used in tenured heap after a full garbage collection cycle. A healthy LDS should be at 50% or less when using the Parallel Old garbage collection algorithm. Garbage collection logs should be captured for a period long enough for at least one full garbage collection to occur (when using parallel old garbage collection). In addition, memory problems often present themselves as periods of high CPU utilization where the system appears to become unresponsive. Cluster instability is another frequent symptom.
  - Database locks are not occurring.
  - WebLogic Server thread locks are not occurring.
- The I/O capacity of the database storage sub-system and of the shared WebLogic Server persistent store has been validated.

Useful information to share:

- What are the average and maximum OSM order creation volume targets per day and during the busy hour?
- If orders coming into OSM (for example, COM) generate additional orders (for example, SOM), what is the expected generated order volume for each?
- What is the expected line item count for average size orders and the expected line item count for large size orders? Indicate the approximate percentage of large orders.
- Indicate the approximate percentage of expected revision orders, if applicable.
- Indicate the average order duration time (number of days to complete an order).
- Indicate the average number of cartridge types and versions planned to be deployed on the production system.
- Describe any bulk operations that may occur during the day including the number of orders submitted and at what time they will be processed.
- Indicate whether Oracle has previously provided hardware sizing recommendations for the deployment.
- Share a summary of performance test results that specifies how the production system (for example, the application servers, the database, and the storage) responds (for example, the CPU, memory, and I/O) during the target peak load with the target number of cartridge versions deployed. This provides a baseline for comparison when investigating any future performance-related issue.

Checking for a Migration Strategy and Production Schedule

Many OSM projects require a migration phase. Some options include an in-place upgrade of an existing OSM database, or setting up a new parallel instance of OSM and cutting over all orders to the new instance at one time or progressively (for example, blocks of subscribers).

Checklist:

- If an in-place OSM database upgrade is planned.
  - Ensure that completed partitions and orders have been purged to reduce the amount of data to be upgraded.
  - Ensure that the procedure has been tested with a full production database and includes a back-out strategy.
- The schedule for migration, going into production, and key post production activities is defined and Oracle Support is notified.

Useful information to share:

- What is the planned migration procedure, including the back-out procedure?
- What is the schedule for going into production and post going into production activities?

Checking for Database Management Procedures

Determining partition size and deciding on a purging policy are key to ensure that storage capacity is reclaimed in a timely fashion and system performance maintained. For more information, see the discussion about managing the OSM schema in *OSM System Administrator's Guide*.

Checklist:

- Oracle recommends that OSM uses a partitioned database schema to benefit from the ability to purge partitions and from other partition maintenance procedures.
- Ensure that the average expected daily and weekly storage consumption rate has been measured.
- Ensure that a procedure exists for regularly creating empty partitions before they are needed. This procedures should align with the expected rate at which the partitions are consumed. This procedure is typically performed after completed partitions are purged.
- Ensure that you have defined a purging strategy: partition based purging, row based purging, or a hybrid of both. For more information, see *OSM System Administrator's Guide*.
- Ensure that there is a schedule for backing up the system and collecting statistics and that these activities are included in any performance tests that include purging (for example, during longevity tests).
- If you are using partition based purging, do the following:
  - Ensure that partition size and purge frequency are defined and documented. The purge frequency should provide enough time for most orders (for example, 98%) to complete in partitions to be purged while factoring in the retention period. This ensures that storage capacity is reclaimed in a timely and predictable fashion.
  - Ensure that performance tests are run with purge_partitions, or drop_empty_partitions, or both to ensure that the database, storage and OSM are tuned for maximum purging performance.

  - If you have inter-order dependencies you may need to use special purge criteria for which you should conduct tests and ensure that subsequent order processing of dependent orders works as expected.
  - Ensure that a backup is taken before running the partition purge in case an error occurs during the purge.
  - Ensure that optimizer statistics are gathered after running a partition purge so that the updated statistics take into account the consolidated partitions.
  - Ensure that the database administrator is trained to recover from common purging problems identified in the discussion about troubleshooting and error handling in *OSM System Administrator's Guide*.

- If you are using row based purging, do the following:
  - Ensure that purge frequency is defined and documented. The daily or weekly purge frequency is primarily a function of the amount of CPU and I/O available throughout the day and week. You must run performance tests to determine whether you have enough CPU and I/O to meet the expected daily and weekly order volume.
  - Ensure that partition size is large enough that space freed up in a partition can be reused by new orders in the same partition. Generally partition size should be able to accommodate at least two months worth of order data.
  - Plan a schedule for when row based purging will happen on a daily and weekly basis and at what parallelism this purging will use. If you have high order processing volume, you would use less parallel processing. If you have less order processing volume, you can use more parallel processing.
- Ensures that the maximum expected number of OSM partitions multiplied by the number of sub-partitions (for example, 32) never exceeds 4800. This is the practical database server scalability limits on number of partitions.
- Ensure that enough storage capacity is available to persist all in-progress orders, pre-created empty partitions including a margin of safety in case of delays in purging completed partitions.
- If the OSM solution includes additional database tables, ensure that you have database management strategies that include purging (if applicable) and backup.
- Ensure that operational procedures exist for purging and dropping partitions, for undeploying unneeded cartridge versions, and for creating empty partitions to receive new orders for the upcoming production period (for example, the next month).

Useful information to share:

- What is the partition size (number of orders), number of Oracle RAC nodes, and number of subpartitions configured?
- If using partition based purging, what is the expected number of days (starting from first order creation) required for a partition to complete most (for example, ~98%) of its orders to be ready for purging?
- If using row based purging, what is the row based purging schedule and purge rate (purged orders per minute).
- How long do you plan to retain orders (order retention policy) after orders have completed?
- How frequently do you plan to add and drop partitions?
- Are you planning to use purge_partitions, to retain a small percentage of orders still not completed, or to drop_empty_partitions, having completed (or aborted) all orders?

Checking for Database Optimizer Statistics Schedule

Because OSM performance depends on optimal database performance, you must properly gather database optimizer statistics on a daily basis. For more information about database optimizer statistics, see *OSM System Administrator's Guide*.

Checklist:

- Ensure that the daily and weekly OSM production schedule is defined, including expected peak order processing hours and the time when OSM batch operations occur.
- Ensure that a job is scheduled to gather database statistics for highly volatile OSM order tables during expected daily high-volume periods. Incremental statistics are disabled for these tables.
- Ensure that a strategy is defined for handling low to medium volatility OSM order tables according to your scenario and that the tables are included in the daily statistics gathering job if applicable.
- Enable incremental statistics for other OSM order tables and ensure that enough low-volume time exists in the day for the database to gather statistics for these tables automatically. Ensure that you confirm when this normally occurs.
- If batch OSM operations do not naturally occur before database statistics are automatically gathered daily, you may need to schedule a job to gather statistics explicitly.
- Ensure that a backup and disaster recovery strategy is in place for this OSM deployment.

Useful information to share:

- What is the daily OSM production schedule (peaks, batch operations, backups, and database statistics gathering)?
- What are the key elements of the OSM database statistics gathering procedure?

Checking for Outage and Order Failure Plans

It must be possible to stop incoming orders, to be queued in an upstream system (for example, Oracle AIA), for a planned OSM outage or an unplanned incident.

Ensure that a strategy exists to identify and correct order-related failures.

- Ensure that operational procedures exist for stopping the creation of new orders in OSM and for dealing with systems that may be sending responses to OSM.
- Ensure that procedures exist for introducing queued orders into OSM in a gradual way (see OSM Warm-up Procedures in 1919049.1) when needed (for example, after major changes to the system).
- Ensure that a strategy, procedures, and tools exist to manage orders in fallout, including orders that may be stuck. This includes procedures for dealing with JMS messages that go into fallout queues and that may need to be resubmitted into processing queues.

Checking for Change Control Management Plans

As for any mission-critical software, when OSM is in service you must make changes to the environment (for example, configuration changes or cartridge deployments) in a controlled manner. You must ensure the existence of detailed steps for the introduction of any changes.

Provide the following:

- Document and test a specific backup strategy so that if anything that was planned fails, a reversal of the changes can be performed to bring the system back to its previous state. This would potentially need to happen at the database level, at the solution level, at the cartridges level, or even at the core OSM application level.
- Document of potentially involved and affected systems.
- Document those that implemented the changes (including prerequisites), their roles and how to contact them during the time period the changes are scheduled for.
- Document the approximate duration for every step.
- Document prerequisite steps to be performed on OSM and external systems with outcomes and responsibilities, for example:
  - Taking a snapshot or backup of various other systems in case rollback is required.
  - Stopping the upstream flow of orders into OSM.
  - Waiting until the OSM inbound queues are drained and all orders are started.
  - Stopping the various managed servers.
  - And so on.
- Document the step-by-step plan for what is be done on OSM and external systems with outcomes and responsibilities.
- Document the steps that must be performed after the change with outcomes and responsibilities.
- Document validation steps to ensure that the changes are active and working as expected.
- Document any hardware, operating system, database, application server, and OSM changes.

In the event of a serious production issue, Oracle support may request change control documentation to understand whether any recent changes may have contributed to a production issue. The availability of this information can greatly shorten the resolution of an issue. It's also important to retain the ability to test the system under volume after going into production, which requires a separate pre-production environment.

Checklist:

- Ensure that a change control management strategy exists that defines how changes are tracked, applied one at a time (or in small groups), validated, approved, and monitored after implementation (in case of problems).
- Ensure that an initial baseline of configuration is captured and change control documentation exists.
- Ensure that procedures exist (and are tested) for introducing solution cartridge changes. The procedure should include how to version cartridge changes, a regression test strategy and plan, and the ability to validate in-progress order compatibility if un-versioned cartridge changes are planned (see 1612273.1)
- Ensure that procedures exist for applying patches (OSM, WebLogic Server, and database), including a test strategy.
- Ensure that procedures exist for gathering database statistics after major changes.
- Ensure that a pre-production environment exists to test future changes and investigate product issues (including performance issues) that may occur.

Useful information to share: Describe the pre-production environment and how it compares (for example, processing capacity) to the production environment. Is it enough to validate performance volume related issues or changes?

Checking for Performance Monitoring Procedures

It is important to monitor the OSM workload and key system metrics to quickly detect any changes in workload or performance characteristics. Oracle support may request performance monitoring information to investigate any performance-related issue that may arise in production.

- Ensure that a method exists for capturing and tracking the daily and hourly volume of OSM orders created and completed and the volume of tasks run.
- Ensure that a method exists for capturing and tracking the creation and completion of large orders (if applicable).
- Capture and track key system performance metrics such as application server and database server CPU utilization, memory consumption, and storage I/O and capacity.

B

Upgrading OSM to an Oracle RAC Environment

This appendix describes how to upgrade an existing Oracle Communications Order and Service Management (OSM) system that uses a single-instance database to an environment that uses Oracle Real Application Clusters (Oracle RAC). You can perform the upgrade using one of two methods:

- Upgrade OSM after converting the database to Oracle RAC
- Upgrade OSM to Oracle RAC using Data Pump Import Export

Upgrading OSM After Converting the Database to Oracle RAC

Using this method, you convert your single-instance database to an Oracle RAC database first, and then run the OSM installer to upgrade the OSM database schema and configure the WebLogic Server resources.

See *Oracle Real Application Clusters Installation Guide for Linux and UNIX* for database conversion instructions.

See "Performing the OSM Application Upgrade " for detailed installation steps.

Upgrading OSM to Oracle RAC Using Data Pump Import and Export

If it is not feasible to upgrade the database instance due to circumstances in your environment, such as the presence of other applications on the instance, you can export the data from the existing database instance and import it into the new version. The recommended method for doing this is Oracle Data Pump Import and Export. After the data is imported, you can run the OSM installer.

Upgrade Overview

To upgrade OSM to Oracle RAC using Oracle Data Pump, perform the following general steps:

1. Shut down the OSM Server. See "Shutting Down the OSM Server."
2. Turn off the OSM Notification Engine. See "Turning Off the Notification Engine."
3. Export and then import the data from the database using Oracle Data Pump. See "Exporting and Importing the Database Data."

> **Note**
>
> Step 3 should be performed by a Database Administrator. Steps 2 and 5 may need to be performed by someone with database administration experience.

4. Run the OSM installer to upgrade the OSM database schema and configure the WebLogic Server resources. See "Performing the OSM Application Upgrade " for detailed installation steps.
5. Restart the Notification Engine. See "Restarting the Notification Engine."
6. Restart the OSM Server. See "Restarting the OSM Server."

Shutting Down the OSM Server

Stop OSM according to the instructions in *OSM System Administrator's Guide.*

Turning Off the Notification Engine

Prior to migration, ensure that the database is not processing any jobs. You do this by manually turning off the Notification Engine.

1. Log in to SQL*Plus as the primary OSM schema user (not the Rule Engine user).
2. Enter the following commands into SQL*Plus:

```
update om_parameter
   set value  = 'N'
   where mnemonic = 'run_jobs';
update om_parameter
   set value  = '86400'
   where mnemonic = 'job_monitor_interval';
commit;
```

   The database will stop running new jobs.
3. Enter the following commands into SQL*Plus:

```
begin
   for i in (select job from user_jobs) loop
      dbms_job.broken(i.job, TRUE);
      dbms_job.remove(i.job);
   end loop;
   commit;
end;
```

   The database will halt any jobs that are currently running.
4. Enter the following command into SQL*Plus:

```
select * from user_jobs;
```

   This should return no jobs running. If there are still any jobs running, consult a DBA.

Exporting and Importing the Database Data

Export the data from the non-Oracle-RAC database and import it to the Oracle RAC database. The recommended method for exporting and importing the data from the database is to use the Oracle Data Pump Import and Export utility.

For information about how to use Oracle Data Pump, see the Oracle Database documentation. Following are a few considerations for OSM:

- It is best to export and import in schema mode (the default).
- Ensure that you export and import both the OSM Primary Schema and the OSM Rule Engine Schema.

- By default, Data Pump Import creates the schema users and performs the necessary grants. It is a good idea to use the default, and if you do, you must ensure that users with the same names do not already exist in the new database instance.

Running the OSM Installer

See "Performing the OSM Application Upgrade " for detailed installation steps.

Restarting the Notification Engine

After you have migrated the OSM database to the new environment, you can restart the Notification Engine.

1. Log in to SQL*Plus as the primary OSM schema user (not the Rule Engine user).
2. Enter the following command into SQL*Plus:

   ```
   select * from user_jobs;
   ```

   If this command returns two running jobs, you do not have to restart the Notification Engine manually: go to "Restarting the OSM Server."
3. Enter the following commands into SQL*Plus:

   ```
   update om_parameter
      set value  = 'Y'
      where mnemonic = 'run_jobs';
   update om_parameter
      set value  = '600'
      where mnemonic = 'job_monitor_interval';
   commit;
   ```

   The database will start running new jobs.
4. Enter the following commands into SQL*Plus:

   ```
   begin
      om_job_pkg.resubmit_jobs(om_const_pkg.v_rule_task_type, 1, 10, 0, 0, 0, 0);
   end;
   ```

   The database will start the appropriate jobs.

Restarting the OSM Server

Start OSM according to the instructions in *OSM System Administrator's Guide.*

C

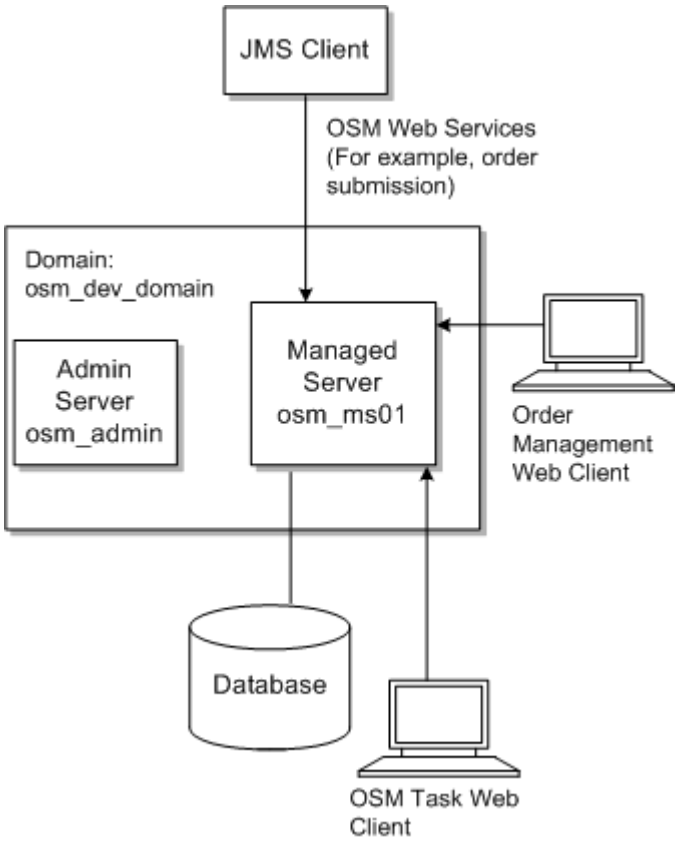
OSM Development System Guidelines and Best Practices

This appendix provides guidelines and best practices for the various technologies and components that make an Oracle Communications Order and Service Management (OSM) development system.

OSM Development Planning Overview

Figure C-1 shows a development system topology with one administration server, one managed server at the application server layer. The database layer contains a non-partitioned database. This OSM system can be deployed on a single Linux machine or VM.

Figure C-1 OSM Development System Topology



Installing OSM Components on a Windows System

The OSM Server is not supported on Windows systems.

If you plan to use the Design Studio component of Oracle Communications Service Catalog and Design on a Windows system, you should download the SDK for your version of OSM and unzip it on the Windows system. If you plan to generate reports using the command line utility of the OSM Reporting Interface, download the SDK for your version of OSM and unzip it.

Hardware Requirements for Development Systems

A small development installation can be installed on a system meeting the following minimum requirements:

- Two GB RAM
- One dual-core processor
- 10 GB of disk space

The numbers provided are guidelines only. Actual disk and memory usage will vary based on the number of users, amount of data, and transaction volume.

For a small, low-volume installation, you can colocate components such as the Oracle Database and WebLogic Server on the same computer. In this case, the minimum RAM required is four GB.

Preparing the Database

The following sections provide information about installing and configuring the Oracle database.

Oracle Database Kernel Configuration

If Oracle Database is installed on the same server as OSM, you must ensure that the appropriate kernel configuration for the database is present.

Downloading and Installing the Oracle Database

For information about installing Oracle Database, see the Oracle Database installation documentation.

You can install OSM with Oracle Database Release 12*c* pluggable databases (PDB) within a multitenant container database (CDB) or within a non-container database.

In order to set all of the options in the database, you may not want to create a database during the database software installation. It may be more convenient to run the Oracle Database Configuration Assistant (DBCA) after installing the database software. See *Oracle Database 2 Day DBA* for more information about creating a database with DBCA.

After you install the database, you must add a permission to the database administrator user that you are going to use during the OSM installation. To do this, log in to SQL*Plus as sysdba and run the following:

```
grant create any context to sysuser as sysdba with admin option
```

where *sysuser* is a user with sysdba privileges that you intend to use during OSM installation.

Database Configuration Considerations for Development Instances

Use local listeners. With this option, each Oracle RAC instance is configured to register only with its listener in the same physical server.

The initial extent size for partitioned tables is set to 8MB. If you have many sub-partitions, a large amount of space can be allocated. For example, a table with 64 sub-partitions will be allocated with an initial space requirement of 512MB. Although this is not an issue for production environments, it can become an issue in development or testing environments with limited storage capacity.

To minimize the database space consumed by OSM:

- Set the **deferred_segment_creation** initialization parameter to **True** (the default value.)
- Install OSM with no partitions, or a small number of partitions (for example, 4.)

Database Parameters

Use the values of the database parameters specified for production systems.

Tablespaces

For non-production environments, one permanent tablespace is enough for the OSM data, and OSM can use the default temporary tablespace for the database instance.

Preparing WebLogic Server

The following sections provide information about installing and configuring the WebLogic Server.

Installing WebLogic Server Software

The software for WebLogic Server and Application Development Framework (ADF) is included in the OSM software media pack. You download the OSM software media pack from the Oracle software delivery website:

https://edelivery.oracle.com/

You install WebLogic Server on all machines that will participate in your domain. The installation directories must be the same on all machines. For complete installation instructions and general information about installing and configuring WebLogic Server, see the WebLogic Server documentation.

> **Note**
>
> See "*OSM Compatibility Matrix*" for WebLogic Server version and patch information. Ensure that you use the WebLogic Server documentation specific to the required WebLogic Server version.

WebLogic Server Software Installation Overview

To install WebLogic Server:

1. Ensure that you have installed the version of Java and the Java Development Kit (JDK) that is supported by OSM, not the version included with WebLogic Server. See "*OSM Compatibility Matrix*" for more information. If you are using a 64-bit operating system, Oracle recommends using a 64-bit JDK with the OSM WebLogic server instance to increase performance.
2. Set environment variables for the version of Java that is supported by OSM, not the version included with WebLogic Server. Do the following:
   - Set JAVA_HOME to the location of the supported Java version.
   - On a UNIX system, add **$JAVA_HOME/bin** to the PATH variable.
3. Install the WebLogic Server software as described in *Oracle Fusion Middleware Installing and Configuring Oracle WebLogic Server and Coherence*. When prompted for the installation type, select **Complete**.
4. Download and install any necessary patches from Oracle support. Follow the instructions in the **README.txt** file that is included with the patch.
5. Create database schemas. See "Creating Database Schemas Using RCU".
6. Create a WebLogic Server domain.
7. Configure the WebLogic Server domain.

Creating Database Schemas Using RCU

After you install the WebLogic Server software, create schemas in the database. You create the schemas using the Repository Creation Utility (RCU), which is included in the WebLogic Server installation.

The schemas are required for creating the WebLogic Server domain with the JRF template. Each schema can be used by only one domain. If you create a new domain, you must also create new schemas.

Before creating the schemas, ensure that you have your database connection string, port, administrator credentials, and service name ready.

To create database schemas using RCU:

1. Go to the *Middleware_home***/oracle_common/bin/** directory.
2. Run RCU using the following command on UNIX:

   ```
   ./rcu
   ```

   The Repository Creation Utility Welcome screen appears.
3. Click **Next**.

   The Create Repository screen is displayed.
4. Select **Create Repository**, and then click **Next**.

   The Database Connection Details screen is displayed.
5. Enter database details in the fields provided (for example, Database Type, Host Name, Port, and so on), and then click **Next**. Use the host name of one of the Oracle RAC instances. Do not use the SCAN IP address.

The installer checks the prerequisites. After the prerequisite checks are completed, click **OK**, and then click **Next**.

The Components screen is displayed.

6. Select the **Create a New Prefix** option, and enter a new prefix for the schema that will be used for the WebLogic domain.
7. In the table, expand the **AS Common Schemas** component, and select the following components:
   - Oracle Platform Security Services (*prefix*_OPSS)
   - Audit Services (*prefix*_IAU)
   - Audit Services Append (*prefix*_IAU_APPEND)
   - Audit Services Viewer (*prefix*_IAU_VIEWER)
   - Common Infrastructure Services (*prefix*_STB)

   where *prefix* is the prefix name you entered at step 6.
8. On the Schema Passwords screen, select the **Use the same password for all schemas** option. Enter and confirm a password to use for the schemas, and then click **Next**.

   The Map Tablespaces screen is displayed. You can click the **Manage Tablespaces** button when you want to change existing tablespaces. For this procedure, click **Next**. A Repository Creation notification is displayed. Click **OK**. Tablespaces are created, and the progress is displayed in a pop-up notification. When the operation is completed, click **OK**.

   The Summary screen is displayed.
9. Click **Create**.

> **Note**
>
> RCU may take several minutes to create the schemas. If creating the schemas takes an unusual amount of time, Oracle recommends that you purge the database recycle bin to ensure that RCU schemas are created more quickly the next time. For more information, see "OSM and RCU Installers Are Slow to Run Database Tablespace Query".

Creating the WebLogic Server Domain

To create the WebLogic Server domain for OSM:

1. Create a domain as described in *Oracle Fusion Middleware Creating WebLogic Domains Using the Configuration Wizard*, and make the following selections:
   - On the Templates screen:
     - Select the **Oracle JRF** and **WebLogic Coherence Cluster Extension** templates.
     - Select the **Oracle Enterprise Manager** template if you want to use Oracle Enterprise Manager Fusion Middleware Control to view and manage OSM logs.
     - The **Basic WebLogic Server Domain** template is selected by default and you cannot deselect it.
   - On the Domain Mode and JDK screen, select or browse to the version of the Java Development Kit (JDK) that is supported by OSM. See "*OSM Compatibility Matrix*" for more information.

- On the Database Configuration Type screen, configure the schemas that the system requires, which you created in "Creating Database Schemas Using RCU". Enter the data, including the schema owner that you created, and then click **Get RCU Configuration**. Click **Next**.
- On the Component Datasources screen, verify that the component schemas are configured and then click **Next**.
- On the Advanced Configuration screen, do one of the following:
  - To create an administration server without any managed servers, select only **Administration Server**.

    This configuration is appropriate for most development and test environments. However, because performance testing and staging environments should use the same server configuration as the production environment, a different configuration may be more appropriate.
  - To create an administration server with a single managed server select at least **Administration Server**, **Managed Servers, Clusters and Coherence**, and **Deployments and Services**.

    This configuration is appropriate for production environments that are not highly available. Having the administration traffic on the administration server and the application traffic on the managed server ensures that critical administration operations (such as starting and stopping servers, changing a server's configuration, and deploying applications) do not compete with high-volume application traffic on the same network connection.
  - To create an administration server with multiple managed servers, select at least **Administration Server**, **Managed Servers, Clusters and Coherence**, and **Deployments and Services**.

    This configuration, also called a cluster, is appropriate for all environments that need high availability. If you are using Oracle RAC, you must configure the same number or more of managed servers in the cluster as there are Oracle RAC node instances. For example, if you have two Oracle RAC nodes, you must have at least two managed servers.

    See "Installing and Configuring the WebLogic Server Cluster" for detailed instructions about creating a WebLogic Server cluster.
- If you are creating an administration server with a single managed server, do the following:
  - On the Managed Servers screen, add one managed server.
  - On the Deployments Target screen, target all the deployments to the managed server. The targets you select must have ADF installed.

Increasing the Memory Settings for WebLogic Servers

OSM requires more memory than the amount configured for the WebLogic server by default. The values provided in the following procedure are appropriate for most development environments, but the memory available in your environment may vary.

To increase the memory settings for non-clustered WebLogic servers on UNIX and Linux:

1. Back up the *domain_home***/bin/setDomainEnv.sh** file by saving a copy with a different name.
2. Open the *domain_home***/bin/setDomainEnv.sh** file in a text editor.

3. Search for the following:

   ```
   USER_MEM_ARGS="
   ```

4. Do one of the following:

   - If you find the search text, change the value of the variable so that the following options are set to the following options or larger:

     ```
     -Xms4g -Xmx4g -Xmn2g
     ```

   - If you do not find the search text, do the following:

     a. Search for the following line:

        ```
        # IF USER_MEM_ARGS the environment variable is set, use it to override ALL
        MEM_ARGS values
        ```

     b. Above the line that you searched for, add the USER_MEM_ARGS environment variable as follows:

        ```
        USER_MEM_ARGS="-Dweblogic.wsee.useRequestHost=true -Xms4g -Xmx4g -Xmn2g"

        # IF USER_MEM_ARGS the environment variable is set, use it to override ALL
        MEM_ARGS values
        ...
        ```

5. Save and close the file.

Configuring the WebLogic Server Domain

The following tasks can be performed either before or after installing OSM, but Oracle recommends performing them before installing OSM.

Preventing Connection Timeout when Using a Remote Database

To prevent a connection timeout when your database and server are in separate locations, or when deploying large cartridges, do the following before deploying cartridges:

1. Increase the value of the Stuck Thread Max time parameter as follows:

   a. Log in to the WebLogic Remote Console.

   b. In the **Edit Tree**, expand **Environment** . From **Environment**, click **Servers**

      The Summary of Servers page is displayed.

   c. Click the name of the WebLogic server where you want to deploy the cartridges.

      The configuration parameters for the server are displayed on a tabbed page.

   d. Select the **Advanced Tab**. From **Advanced Tab**, select the **Tuning** tab and modify the value of the **Stuck Thread Timer Interval** parameter to 1200 in seconds or greater.

   e. Click **Save**.

2. Increase the value of the Timeout Seconds Java Transaction API parameter as follows:

   a. In the **Edit Tree**, select **Services** of the WebLogic Remote Console.

      The configuration parameters for the domain are displayed on a tabbed page.

   b. Click the **JTA** tab, and modify the value of the **Timeout Seconds** parameter to an appropriate value. In most cases, a value of 600 seconds is enough.

> **Note**
>
> If the value is less than 600 when you run the Installer, you are prompted to increase it during installation.

c. Click **Save**.

3. Increase the value of the cartridge deployment transaction timeout parameter as follows:

   a. Open the **startWebLogic.sh** (UNIX or Linux) file for your WebLogic domain.

   b. Search for the following string:

   ```
   -
   Dcom.mslv.oms.cartridgemgmt.DeployCartridgeMDB.CartridgeDeploymentTransactionTime
   out
   ```

   c. Increase the value of the parameter.

   For example, in **startWebLogic.sh** in a UNIX or Linux environment:

   ```
   export JAVA_OPTIONS="${JAVA_OPTIONS} -
   Dcom.mslv.oms.cartridgemgmt.DeployCartridgeMDB.CartridgeDeploymentTransactionTime
   out=600"
   ```

   For example, in **startWebLogic.cmd** in a Windows environment:

   ```
   set JAVA_OPTIONS=%JAVA_OPTIONS% -
   Dcom.mslv.oms.cartridgemgmt.DeployCartridgeMDB.CartridgeDeploymentTransactionTime
   out=
   ```

   > **Note**
   >
   > The maximum value for this parameter is 3600 seconds. If you exceed this limit, the value is restored to the default of 600 seconds. Try again with a value that is less than or equal to the maximum.

   d. Save and close the file.

See the WebLogic Server documentation for more information about these parameters.

Other Supported High-Availability Options

The following section provides other high-availability options for OSM. These options are different than those described in the other chapters of this book.

Configuring Oracle Database with Clusterware

Oracle Clusterware, when installed on servers running the same operating system, enables the servers to be bound together to operate as if they are one server, and manages the availability of applications and Oracle databases.

With Oracle Clusterware, you can provide a cold cluster failover (also known as cold standby) to protect an Oracle instance from a system or server failure. The basic function of a cold cluster failover is to monitor a database instance running on a server and, if a failure is detected, to restart the instance on a spare server. Network addresses are failed over to the backup node. Clients on the network experience a period of lockout while the failover occurs and are then served by the other database instance after the instance has started.

Setting Up the Database and Clusterware for Cold Standby

A typical cold standby configuration consists of Oracle Clusterware and two Oracle single-instance databases running on separate physical servers with shared disk storage. In this configuration, when node A fails, the database, listener, and ASM instance automatically fail over to node B.

The following procedure describes how to set up Oracle Database and Clusterware to provide cold failover for a single-instance, non-cluster, Oracle database.

1. Install Oracle Clusterware on two physical machines.
   - For Linux, see *Oracle Grid Infrastructure Installation Guide for Linux*.
2. Install a clustered ASM (Automatic Storage Management) *home* and instance.

> **Note**
>
> Oracle recommends the use of ASM to optimize storage performance and usage, and to tolerate storage failures. You may need to use other storage systems, such as Oracle Automatic Cluster File System (ACFS), or storage on raw devices. For more information, see the respective product documentation.

3. Create an ASM disk group on shared storage which is accessible to both nodes A and B.
4. Install a software-only version of Oracle Database into a new Oracle home on both nodes.
5. Use the database configuration assistant (DBCA) to create a single-instance database on node A, sharing data files stored on the ASM disk group created in step 3.
6. Create a new virtual IP address on node A.
7. Implement the scripts provided in the Oracle white paper *Using Oracle Clusterware to Protect A Single Instance Oracle Database*. The scripts must be run on both nodes to protect the single-instance database, virtual IP address, and listener.
8. Start the database, listener, and ASM instance on node A using Oracle Clusterware.
9. Start the ASM instance on node B.

You have now configured your database instances for cold cluster failover. Once you have installed OSM, you can configure a WebLogic instance to automatically restart when the database fails. See "Configuring WebLogic for Cold Cluster Failover" for more information.

Configuring WebLogic for Cold Cluster Failover

This section describes how to configure a WebLogic instance to restart automatically when a database configured for cold cluster failover fails. For each server in the cluster, navigate to the **Health Monitoring** tab and set the following parameters:

- **Auto Kill if Failed** - Select this parameter to enable Node Manager to automatically stop the server if its health state is failed.
- **Auto Restart** - Select this parameter to enable Node Manager to automatically restart the failed server.
- **Restart Delay Seconds** - Set this value to the number of seconds Node Manager should wait before attempting to restart the server. The value should allow ample time for the database failover to complete, for example, 300 seconds.

You may also want to consider increasing the **JMS Redelivery Limit** and possibly the **JMS Redelivery Delay Override** parameters to ensure that the JMS message redelivery limit is not exceeded during database failover. If the redelivery limit for a JMS message is exceeded, the message is normally delivered to a fallout queue and the symptom is a stuck order. See *OSM System Administrator's Guide* for more information about JMS redelivery configuration settings.

Oracle RAC Active-Passive

OSM supports Oracle RAC in active-passive configuration. Oracle recommends using an active-active configuration for maximum availability.

In active-passive Oracle RAC, an active instance handles requests and a passive instance is on standby. When the active instance fails, an agent shuts down the active instance completely, brings up the passive instance, and application services can successfully resume processing. As a result, the active-passive roles are now switched.

Cold Cluster Failover

Cold cluster failover consists of Oracle Clusterware and two or more Oracle single-instance databases running on separate physical servers sharing disk storage. Oracle Clusterware monitors the primary active database instance and provides the capability of failover to a cold standby database in case of failure, thus ensuring high availability. Clients on the network experience a period of lockout while the failover occurs and are then served by the other database instance after the instance has started.

D

OSM Installer Properties

This appendix lists and describes the configuration properties for Oracle Communications Order and Service Management (OSM).

> **Note**
>
> All passwords are encrypted and if you want to directly change the properties file, you can leverage the tool that is included. This will allow you to encrypt a plain-text password and copy-paste the encrypted value into this file.

WebLogic Parameters

Table D-1 lists the WebLogic parameters for installing OSM.

Table D-1 WebLogic Parameters for Installing OSM

Parameter Name	Description
capacity_increase	JDBC connection pool increase size.
ctrl_coherence_cluster_for_cleanup	Values of existing custom orosm coherence clusternames. This is a read only parameter.
ctrl_coherence_cluster_to_create	Coherence cluster name. This is a read only parameter.
ctrl_ds_list_for_cleanup	Give the list of data source **existing_wls_ds_replaceurces** to be cleaned up. This is a read only parameter.
datasource_x_config	Provide the additional RAC database connection information if **use_oracle_rac** is *true* and **rac_config** is *now.* Example JSON: `[{ "dataSourcePoolName": "PoolName", "target": "target", "dbUrl": "dbURL", "jndiName": "jndiName", "dataSource": datasource, "dataSourceType": "typeof datasource" }, { "dataSourcePoolName": "PoolName1", "target": "target1", "dbUrl": "dbURL1", "jndiName": "jndiName1", "dataSource": datasource1, "dataSourceType": "typeof datasource1" }]`

Table D-1 (Cont.) WebLogic Parameters for Installing OSM

Parameter Name	Description
deployment_target_type	WebLogic deployment target type: • If target type is cluster then give *cluster* • If target type is standalone adminserver then give *adminserver* • If target type is standalone managedserver then give *managedserver*
existing_wls_ds_list	Lists the existing Datasources in list format: For example: `[ { "name": "nameofDS", "is_rac": "true/false", "host": "hostdetails", "port": "port", "service": "service", "sid": "sid info", "user": "userdetails" } ]`
existing_wls_ds_replace	Set to *true* incase of upgrade if existing datasource needs to be replaced with new datasource. The default value is *true*
front_end_host	The listen address of the HTTP proxy server for the WebLogic cluster.
front_end_http_port	The HTTP listen port of the HTTP proxy server.
front_end_https_port	The HTTPS listen port of the HTTP proxy server.
init_capacity	JDBC connection pool initial capacity.
jms_store	JMS store type Example: jdbc
managed_servers	Managed server address list when deployment target type is cluster. Example JSON: `[ { "name": "cluster1", "url": "cluster-dev" }, { "name": "ms1", "url": "ms1-dev" } ]`
max_capacity	JDBC connection pool max capacity.
oms-automation	Internal user used by OSM.
oms-internal	Internal user used by OSM.
oms-metrics	Internal user used by OSM.
osm_app_deployed	Set to *true* in case of Upgrade otherwise set to *false*.

Table D-1 (Cont.) WebLogic Parameters for Installing OSM

Parameter Name	Description
rac_config	Set to *now* if you want to add Oracle RAC database otherwise, set to *later* to configure it manually after installation.
rac_operation_mode	Set RAC Datasource operation modes: • If deployment target is cluster and DB partition is enabled then use *load_balance* or *failover*. • Otherwise use *failover*.
unicast_port	Unicast port number of the coherence cluster.
use_oracle_rac	Set to *true* to configure an Oracle RAC database or *false* to not configure an Oracle RAC database.
weblogic_admin_server_host	WebLogic host name or IP Address.
weblogic_admin_server_port	WebLogic administration server port number.
weblogic_admin_user_name	Weblogic Admin user name.
weblogic_admin_user_password	Weblogic Admin user Password. All passwords are encrypted.
weblogic_deployment_target	Weblogic Cluster Name.
weblogic_plugin_enabled	Set to *true* in case of SSL enabled.
weblogic_ssl_enabled	Set to *true* to use an SSL connection to the WebLogic Administration Server or *false* if not to use an SSL connection.
well_known_address	The WKA member's server address.

OSM Parameters

Configuring the OSM parameters lets you customize web client behavior and enter specific information in the worklist rows, or the notification e-mail address or any of the nodes you see. Set these parameters depending on your preferences.

Table D-2 lists the OSM parameters for installation of OSM.

Table D-2 OSM Parameters for OSM Installation

Parameter Name	Description
Session-Timeout	Time in minutes that Order Management web client and Task web client sessions remain active.
Server-Domain-Suffix	Domain suffix for the computer(s) on which the OSM server will run.
Support-Cluster	Provides special support for WebLogic clusters when selected.
Attachment-Path	The path used to locate the WebLogic file (T3) service used for storing OSM remark attachments.
Maximum-Attachment-Size	Maximum attachment size, in MB, that can be appended to a remark.
Remark-Change-Timeout	Length of time in hours that a remark can be edited following creation.
Worklist-Rows-Retrieved	Number of rows retrieved when the worklist is refreshed.
Worklist-Rows-per-Page	Number of rows displayed on each worklist page.
Query-Rows-Retrieved	Number of rows retrieved when a query is run.

Table D-2 (Cont.) OSM Parameters for OSM Installation

Parameter Name	Description
Query-Rows-per-Page	Number of rows displayed on each query results page.
Notification-Rows-Retrieved	Number of rows retrieved when the notification is refreshed.
Notification-Rows-per-Page	Number of rows displayed on each notification page.
Notification-Email-Server	DNS name or IP address of your e-mail server.
Notification-Server-Port	Port on which the e-mail server is listening.
Notification-Email-Address	OSM Administrator's e-mail address.
Read-Only-Field-Length	Maximum length of read only fields in the Order Editor.
Event-Buffer-Interval	Interval used to buffer events before sending them to the automation framework.
Landing-Page	The page that by default all Task web client users will be directed to at log in.
Display-Namespace-Version	Set to **1** to show the cartridge namespace version on the Task web client's 'Create Order' page, or set to **0** to not display the cartridge namespace version on this page.
WLS-Server-SSL-Port	SSL Port on which the OSM server is listening.
Task-Processor-Interval	Enter the number of seconds between task processor polls.
Max-Rule-Processor-Count	Enter the maximum number of rule task processors used to evaluate rules.
Max-Delay-Processor-Count	Enter the maximum number of delay task processors used to evaluate delays.
Handler-Factory	For a clustered installation, set this parameter to **com.mslv.oms.handler.cluster.ClusteredHandlerFactory**. For a non-clustered installation, set this parameter to **com.mslv.oms.security.HandlerFactory**.

OSM J2EE Application Properties

The OSM J2EE application properties contain OSM system user settings, UI settings and other application settings. The following table describes each element:

Table D-3 OSM J2EE Application properties

Property Name	Description
osm_admin_email_address	OSM Admin email address
osm_admin_password	OSM admin user encrypted password
osm_admin_username	OSM admin user name
osm_automation_user_password	OSM automation user encrypted password
osm_core_user_password	OSM internal user encrypted password
osm_deploy_admin_password	OSM deploy admin encrypted password
osm_deploy_admin_username	OSM deploy admin user name
osm_landing_page	UI landing page
osm_max_attachment_size	Max attachment size
osm_max_delay_processor_count	Max delay processor count

Table D-3 (Cont.) OSM J2EE Application properties

Property Name	Description
osm_max_rule_count	Max rule count
osm_metrics_user_password	Metrics user password
osm_notification_email_server	Notification email server host
osm_notification_email_server_port	Notification email server port
osm_remark_change_timeout	Timeout setting for checking remark updated
osm_server_domain_suffix	Server domain suffix
osm_session_timeout	session timeout
osm_task_processor_interval	task processor interval
third_party_readme	Set this value to true if you agree to the third party readme text.
undeploy_jdbc_datasource	Set to true in case of OSM Upgrade This is used to upgrade the WLS JDBC resource
undeploy_jdbc_datasource_list	Provide list of JDBC datasources to be deleted in case of OSM Upgrade. This is used to upgrade the WLS JDBC resource

Database Parameters

The database properties contains database admin credential, connection information and OSM schema user and password, partition settings such as count and limit. The following table describes each property:

Table D-4 Database Properties

Property Name	Description
db_admin_password	Sys DBA encrypted Password
db_admin_username	Sys DBA Username
db_default_tablespace	Default tablespace name
db_host	Host name of the database where OSM application is getting deployed/already deployed
db_model_data_tablespace	Model data tablespace name
db_model_index_tablespace	Model index tablespace name
db_order_data_tablespace	Order data tablespace name
db_order_index_tablespace	Order index tablespace name
db_osm	OSM core schema name
db_osm_password	OSM core schema encrypted password
db_partition	Use partition
db_partition_limit	Partition limit
db_partition_size	Partition size
db_port	DB port
db_report	OSM report schema name
db_report_password	OSM report schema encrypted password
db_rule_engine	OSM rule engine schema name
db_rule_engine_password	OSM rule engine schema encrypted password

Table D-4 (Cont.) Database Properties

Property Name	Description
db_service	DB service name
db_subpartition_count	Subpartition count
db_temp_tablespace	Temporary tablespace name
db_timezone_offset_seconds	DB timezone offset
schema_localized	Use localization
undeploy_jdbc_store_list	Provide the list of JDBC/Persistent store names in case of upgrade if you want to delete existing persistent store
undeploy_jms_server_store_map_list	Provide list of jms server(s) in case of upgrade if want to delete existing persistent store

E

Installing OSM on Engineered Systems

This appendix describes recommendations for installing Oracle Communications Order and Service Management (OSM) on engineered systems.

JDBC Recommendations

This section includes recommendations for JDBC.

For engineered systems, Oracle recommends that you use SDP protocol over Infiniband (IB). This protocol enables multiple performance enhancements, such as input/output, thread management, and request handling efficiency. Typical steps to enable the SDP protocol include:

- Ensure the physical Infiniband connection exists and is operational between the WebLogic server host and the database host.
- Set up an SDP listener on the Infiniband network.
- In the JDBC URL, replace TCP protocol with SDP protocol, and if necessary, change the port number to match the SDP listener's port. For example:

  ```
  jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=SDP)(HOST=VIP_addr)(PORT=port))
  (CONNECT_DATA=(SERVICE_NAME=service_name) (INSTANCE_NAME=instance_name)))
  ```

- Manually add the following system properties to the **startWebLogic.sh** or **startManagedServer**_*xx*.**sh**.

  ```
  -Djava.net.preferIPv4Stack=true
  -Doracle.net.SDP=true
  ```

- **Initial Capacity**: Oracle recommends setting this value to Max Capacity for production deployments to avoid shrinking and growing of the JDBC pool size. Setting the Initial Capacity value impacts available resources on both WebLogic server and database server by consuming additional resources at the start up time, and also lengthens the server start up time.
- **Max Capacity**: Set this to a peak and sustainable value, which can be supported by both by WebLogic server (additional memory and processing) and the database server (session resources and concurrency). In uniform cluster deployments of JDBC connection pools to a cluster, the Max Capacity value applies to each WebLogic server node individually, not for the whole cluster. Set Max Capacity such that ((*nodes* x Max Capacity) <= *sessions*), where *nodes* is the number of nodes in the WebLogic Server cluster and *sessions* is the number of peak, concurrent, and safely sustainable sessions to the database server.

> **Note**
>
> Max Capacity is an important parameter that requires iterative tuning based on scenario and workload. One approach is to set it to a high value for peak load tests, and monitor what percentage of it has been used, and then adjust the **MaxCapacity** to at least that high.

- **Statement Cache Size**: The prepared statement cache size is used to keep a cache of compiled SQL statements in memory, thus improving the performance by avoiding repeated processing in the WebLogic server and database. For lightly used data sources, the default value of **10** is enough. For production systems, Oracle recommends a value of 30 to 40.

> **Note**
>
> Each JDBC connection in the Connection Pool creates its own prepared statement cache. When you tune this parameter, consider the additional memory consumption demand caused by (*steady size of Connection Pool* x *Prepared Statement Cache Size*). A demand that is too high may cause "Out of Memory" exceptions on WebLogic server and may disable the connection pool altogether, rendering the server useless. Tuning Statement Cache Size is achieved by an iterative process, influenced by factors of the scenario, workload, and steady state size of the connection pool for the given data source.

SDP does not support Oracle Single Client Access Name (SCAN) addresses. It supports VIP addresses over InfiniBand. Please see the Oracle documentation for more instructions on configuring Engineered Systems with SDP.

Configuring Exalogic

The following sections provide recommendations for configuring the application layer.

Exalogic User Process Limit

Table E-1 shows the user process limit for Exalogic systems.

Table E-1 Exalogic User Process Limits

Parameter	Value
Core file size	0
Data seg size	Unlimited
Scheduling priority	0
File size	Unlimited
Pending signals	774889
Max locked memory	Unlimited
Max memory size	Unlimited
Open files	65536
Pipe size	8
POSIX message queues	819200
Real-time priority	0
Stack size	8192
CPU time	Unlimited
Max user processes	774889
Virtual memory	Unlimited

Table E-1 (Cont.) Exalogic User Process Limits

Parameter	Value
File locks	Unlimited

Exalogic Kernel Parameters

Table E-2 shows the kernel parameters for Exalogic systems.

Table E-2 Exalogic Kernel Parameters

Parameter	Value
net.ipv4.ip_forward	0
net.ipv4.conf.default.rp_filter	2
net.ipv4.conf.default.accept_source_route	0
kernel.sysrq	0
kernel.core_uses_pid	1
net.ipv4.tcp_syncookies	1
kernel.msgmnb	65536
kernel.msgmax	65536
kernel.shmmax	68 719 476 736
kernel.shmall	4 294 967 296
net.ipv4.tcp_rmem	16 777 216 16 777 216 16 777 216
net.ipv4.tcp_wmem	16 777 216 16 777 216 16 777 216
net.ipv4.tcp_mem	16 777 216 16 777 216 16 777 216
net.core.optmem_max	16 777 216
net.core.rmem_max	16 777 216
net.core.wmem_max	16 777 216
net.core.rmem_default	16 777 216
net.ipv4.ip_local_port_range	9000 65500
vm.nr_hugepages	10000
fs.file-max	262144
net.core.netdev_max_backlog	250000
net.ipv4.tcp_timestamps	1
net.ipv4.tcp_window_scaling	1
vm.dirty_background_ratio	3

Table E-2 (Cont.) Exalogic Kernel Parameters

Parameter	Value
vm.min_free_kbytes	1 048 576
net.ipv4.tcp_fin_timeout	15
net.ipv4.tcp_keepalive_time	120
net.core.somaxconn	1024
net.ipv4.tcp_sack	0
net.ipv4.tcp_dsack	0
net.ipv4.tcp_keepalive_probes	3
net.ipv4.tcp_keepalive_intvl	30

OSM WebLogic Server Configuration

Table E-3 shows the WebLogic Server configuration.

Table E-3 OSM WebLogic Server Configuration

Parameter	Value
JTA Timeout	600 seconds
JMS Persistence Mechanism	FileStore on SAN
# of SAF Agents	32: OSM
JDBC: Initial Capacity	64: OSM
JDBC: Increment	1
Statement Cache Size	48: OSM
Work Managers	80: OSM oms.automation.core; 10: oms.xml
Message Bridge Threads	2
Accept backlog	900
FileStore Write Policy	Direct-Write-With-Cache
JDBC: Max Capacity	128: OSM
JDBC: Shrink Frequency	60
JDBC: Row Prefetch Size	200
WLS: Native IO	True
Cluster Messaging Mode	Multicast1

JVM Options

Table E-4 shows the Java virtual machine (JVM) options for an Exalogic system.

Table E-4 Exalogic JVM Options

Parameter	Value
64-bit packages	Yes

Table E-4 (Cont.) Exalogic JVM Options

Parameter	Value
64 bit mode	-d64
Production Mode	True
JVM HotSpot	-server
JVM: net IPv4	-Djava.net.preferIPv4Stack=true
SDP	-Doracle.net.SDP=true
WLS: Sockets	-Dweblogic.SocketReaders=16
WLS: Threads	-Dweblogic.threadpool.MinPoolSize=300

Tuning the Oracle Database

For every Oracle Database installation, there are a number of **init.ora** (or **spfile.ora**) parameters that affect performance. Table E-5 lists important initialization parameters for use with OSM:

Table E-5 Suggested Oracle Database Parameters for Engineered Systems

Parameter Name	Sparc SuperCluster (SSC)	Exadata
db_writer_processes	8	3
filesystemio_options	'setall'	'setall'
processes	5000	5000
sessions	7680	7536
undo_retention	1800	1800
deferred_segment_creation	false	false

Table E-6 suggests initial values for tuning parameters for the Oracle database.

Table E-6 Recommended Initial values of Oracle Database Parameters for Engineered Systems

Parameter Name	Sparc SuperCluster (SSC)	Exadata
memory_target	134217728000	0
open_cursors	15000	1000
pga_aggregate_target	0	8589934592
sga_max_size	77846282240	34359738368
sga_target	77846282240	34359738368

For information about collecting more accurate database statistics, see the knowledge article about best practices for managing optimizer statistics [Doc ID 1662447.1], available from the Oracle support website:

https://support.oracle.com

Configuring Database Schema Partitioning

For detailed information about partitioning the database schema, see *OSM System Administrator's Guide*.

Multi-database Source Configuration Using N Oracle RAC Nodes

Oracle recommends that you use an Active-Active Oracle RAC database configuration to provide scalability and high availability for the database. For more information, see "Oracle RAC Database Active-Active Deployments."

> **Note**
>
> OSM installer sets up only two active Oracle RAC nodes by default. For information about adding more nodes, see "Changing the WebLogic Server or Oracle RAC Database Size".

Database Storage

Recommendations for setting up database storage include the following:

- Use Automatic Storage Management (ASM) for managing data disk storage.
- For storage reliability, use Normal (two-way mirrored) Redundancy and ensure that the tablespace, data files, and redo logs are on this storage.
- Place redo logs, which are sensitive to storage response time, should be put on storage with a service time of less than 5 ms.
- Specify large redo log files to avoid frequent redo log switching. For redundancy, use a mirrored configuration for redo logs.
- Finalize the requirements for latency and IOPS during your hardware sizing exercise.

Recommendations for configuring tablespaces include the following:

- Use Automatic Segment Space Management (ASSM) for each tablespace.
- Whenever disk space permits, use BIGFILE for tablespace creation. This simplifies the management of tablespace allocation by using a single data file for tablespace.
- If you must use a SMALLFILE tablespace, plan for the possibility that a large number of data files might be created for OSM schema. Plan to implement a naming convention for data files and find an ideal location for data files that allows for future growth.

F

Installing OSM on Oracle Cloud Infrastructure

OSM is certified on Oracle Cloud Infrastructure (OCI). The OSM database can be installed on an OCI DB system (BareMetal, Virtual Machine, and Exadata) running version 19c as a regular 1-node database or as a 2-node RAC database. The OSM application, the application server, Java and other middleware components can be installed on an OCI Compute instance (BareMetal and Virtual Machine) running Linux 7.6+. For more information, see the *Installing OSM on Oracle Cloud Infrastructure* (Doc ID: 2609018.1) knowledge article on My Oracle Support. Also, refer to the *Recommended Patches for OSM Software Components* (Doc ID: 2170105.1) knowledge article on My Oracle Support for patch recommendations for both database and middleware components.

> **Note**
>
> At this time, OSM is not supported on Oracle Autonomous Transaction Processing and Oracle Autonomous Data Warehouse.