Oracle® Communications Order and Service Management
System Administrator's Guide
Release 7.2.2
E35414-02

March 2013
## Contents

### Preface

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audience</td>
<td>ix</td>
</tr>
<tr>
<td>Documentation Accessibility</td>
<td>ix</td>
</tr>
</tbody>
</table>

### 1 OSM System Administration Overview

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSM System Administration Tasks</td>
<td>1-1</td>
</tr>
<tr>
<td>About OSM System Administration Programs</td>
<td>1-1</td>
</tr>
<tr>
<td>Directory Placeholders Used in This Guide</td>
<td>1-2</td>
</tr>
</tbody>
</table>

### 2 Starting and Stopping OSM

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Starting and Stopping OSM</td>
<td>2-1</td>
</tr>
<tr>
<td>Starting the OSM Server</td>
<td>2-1</td>
</tr>
<tr>
<td>Verifying that OSM Has Started</td>
<td>2-1</td>
</tr>
<tr>
<td>Stopping OSM</td>
<td>2-2</td>
</tr>
<tr>
<td>Verifying that OSM Has Stopped</td>
<td>2-2</td>
</tr>
</tbody>
</table>

### 3 Setting Up OSM Security

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About OSM Security</td>
<td>3-1</td>
</tr>
<tr>
<td>Adding Users to OSM</td>
<td>3-1</td>
</tr>
<tr>
<td>Adding Users to Groups in the WebLogic Server Console</td>
<td>3-1</td>
</tr>
<tr>
<td>Creating Workgroups as Roles in Design Studio</td>
<td>3-2</td>
</tr>
<tr>
<td>Assigning Users to Workgroups in OSM Administrator</td>
<td>3-3</td>
</tr>
<tr>
<td>Required Groups and Users</td>
<td>3-3</td>
</tr>
<tr>
<td>Using Secure Sockets Layer</td>
<td>3-4</td>
</tr>
<tr>
<td>Changing Secure Sockets Layer Configuration in OSM</td>
<td>3-4</td>
</tr>
<tr>
<td>Setting Up a Caching Realm</td>
<td>3-5</td>
</tr>
<tr>
<td>Using the XML Import/Export Application to Administer Users and Workgroups</td>
<td>3-5</td>
</tr>
<tr>
<td>Using the EncryptPasswords Utility</td>
<td>3-7</td>
</tr>
<tr>
<td>About the EncryptPasswords Utility</td>
<td>3-7</td>
</tr>
<tr>
<td>Running the EncryptPasswords Utility</td>
<td>3-8</td>
</tr>
<tr>
<td>Removing an Encrypted Password</td>
<td>3-8</td>
</tr>
</tbody>
</table>

### 4 Securing Credentials Required to Access External Systems

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About the Credential Store</td>
<td>4-1</td>
</tr>
</tbody>
</table>
### 9 Backing Up and Restoring OSM Files and Data

**About Backing Up and Restoring OSM Files and Data**

- Backup and Restore Overview .............................................................. 9-1
- Backup and Restore Schedule ............................................................... 9-1
- OSM Home Directory ........................................................................... 9-1
- Oracle Database ................................................................................... 9-1
- WebLogic Server Files ........................................................................... 9-2
- Backup and Restore Considerations ..................................................... 9-2

**Backing Up and Restoring the OSM Files**

- Backing Up the OSM Files .................................................................... 9-2
- Restoring the OSM Files ........................................................................ 9-2

**Oracle Database Backup Considerations**

- RMAN Considerations ........................................................................... 9-3
- Oracle Flashback Technology Considerations ........................................ 9-3

**Backing Up and Restoring the WebLogic Server Configuration**

- Backing Up the WebLogic Server Configuration ...................................... 9-3
- Before You Back Up the WebLogic Server ............................................ 9-4
- Setting OSM to Pause Processing of JMS Messages ......................... 9-4
- WebLogic Server Domain Directory ................................................... 9-5
- WebLogic Persistent Store .................................................................... 9-5
- OSM Attachment Directory ................................................................. 9-5
- External Deployments ............................................................................ 9-5
- Remote Managed Servers (Clustered Server Only) ............................... 9-6
- Restoring the WebLogic Server Configuration ....................................... 9-6
- Restoring the WebLogic Server Files .................................................... 9-6
- Setting OSM to Resume Processing of JMS Messages ..................... 9-6

### 10 Monitoring and Managing OSM

**About Monitoring and Managing OSM** ............................................. 10-1

**Accessing the WebLogic Server Administration Console** .................. 10-1

**Using the WebLogic Console to Determine the Status of the OSM Application** ............................. 10-2

**Managing Log Files** ........................................................................... 10-2

- Reading Log Files ................................................................................ 10-2
- Configuring the Log View .................................................................... 10-3
- Log Size and Rotation .......................................................................... 10-3

**Managing Error Message Volume and Logging Levels** ..................... 10-3

- Severity Levels .................................................................................... 10-3
- Displaying Log Levels ........................................................................ 10-4
- Configuring the Severity Levels ......................................................... 10-4
11 Using the XML Import/Export Application

About the XML Import/Export Application ................................................................. 11-1
About Using the XML Import/Export Application ..................................................... 11-1
About XML Import/Export Batch Scripts and Ant Commands .................................... 11-3
   About XML Import/Export Ant Commands and Syntax ........................................ 11-3
   About XML Import/Export Batch Scripts and Syntax ............................................ 11-3
Configuring the XML Import/Export Environment Files ........................................... 11-4
   Configuring the build.properties File for Ant Commands .................................... 11-4
   Configuring the config.bat Script for a Batch Scripts .......................................... 11-5
Configuring the config.xml File XML Import/Export Nodes and Elements ............... 11-6
About Importing and Exporting Metadata ............................................................... 11-8
   About Exporting Metadata ..................................................................................... 11-8
      About the Order of Exported Metadata ............................................................. 11-9
      About Export Layout Options ............................................................................ 11-10
      Keeping the ID Integrity in SQL Rules ............................................................... 11-11
      Configuring and Running an Export ............................................................... 11-12
   About Importing Metadata .................................................................................... 11-13
      Configuring and Running an Import ............................................................... 11-14
      Sample Procedure for Adding a New Workgroup Definition (Role) ................ 11-15
      Sample Procedure for Adding a Task to a Workgroup (Role) .......................... 11-17
About Purging MetaData and Data ........................................................................... 11-19
   Configuring and Running a Database Schema or Cartridge Purge ....................... 11-19
   About Purging Orders ............................................................................................. 11-20
      Deleting In-Flight Orders using SQL Scripts ................................................... 11-20
      Purging Orders with the orderPurge.bat Script on Windows ............................ 11-21
      Running Ant with the orderPurge.xml file On UNIX or Linux to Purge Orders ...... 11-23
About Migrating Orders ............................................................................................ 11-25
   Configuring and Running an Order Migration ................................................... 11-26
About Validating the Metadata Model and Data ....................................................... 11-28
   Configuring and Running an XML Document Validation .................................... 11-28
   Validating an XML Document During the Import or Export Process ................. 11-29
About Creating a Graphical Representation of the Metadata Model ....................... 11-29
   Configuring and Creating a Graphical Representation of a Metadata Model ....... 11-29
   Viewing the Graphical Representation .................................................................. 11-30

12 Configuring Time Zone Settings

Configuring Time Zone Settings ............................................................................... 12-1
13 Troubleshooting OSM

General Checklist for Resolving Problems with OSM .......................................................... 13-1

Diagnosing Some Common Problems with OSM ................................................................. 13-1
  Cannot Log In or Access Certain Functionality ................................................................. 13-1
  System Appears Slow ........................................................................................................ 13-2
  To Avoid Timeout Issues in Design Studio ..................................................................... 13-2
  Error: "Login failed. Please try again." ............................................................................. 13-2
  No Users or Groups Are Displayed .................................................................................. 13-3
  Automation Plug-ins Are Not Getting Called ................................................................. 13-3
  Too Many Open Files ..................................................................................................... 13-3
  Proxy Fails on a Clustered System ................................................................................... 13-3
  Problems Displaying Gantt Charts on Solaris 5.10 Hosted Systems ............................. 13-3
  OSM Fails to Process Orders Because of Metadata Errors .......................................... 13-4
  Error: "No Backend Servers Available" ......................................................................... 13-4
  Quick Fix Button Active During Order Template Conflicts in Design Studio .............. 13-4

Getting Help with OSM Problems ..................................................................................... 13-4
  Before You Contact Support ......................................................................................... 13-4
  Reporting Problems ....................................................................................................... 13-4

14 OSM Log Messages

OSM Catalog Messages ...................................................................................................... 14-1
  Automation Catalog Messages .......................................................................................... 14-21
Preface

This document describes Oracle Communications Order and Service Management (OSM) system administration tasks.

Audience

This document is intended for system administrators, system integrators, Database Administrators (DBA), and other individuals who are responsible for managing OSM and ensuring that the software is operating in the manner required for your business. This guide assumes that users have a working knowledge of Windows XP Professional, Linux, Oracle 11, 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 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.
OSM System Administration Overview

This chapter provides an overview of Oracle Communications Order and Service Management (OSM) system administration tasks.

OSM System Administration Tasks

As an OSM system administrator, you can perform the following tasks:

- Install and configure OSM.
- Monitor and manage OSM. See "Monitoring and Managing OSM" for more information.
- Read OSM log messages. See "OSM Log Messages" for more information.
- Troubleshoot OSM. See "Troubleshooting OSM" for more information.
- Configure the rule engine. See "Configuring the Rule Engine" for more information.
- Set up OSM security. See "Setting Up OSM Security" for more information.
- Partition the OSM Database schema. See "Partitioning Your OSM Database Schema" for more information.
- Import, export, purge, and migrate data and metadata. See "Using the XML Import/Export Application" for more information.

About OSM System Administration Programs

Use the following programs for OSM system administration:

- Use the Oracle WebLogic Server Console to do the following:
  - Start and stop server components
  - Monitor system components
  - Read log files
  - Manage security
- Use the XML Import/Export application to do the following:
  - Export and import the metadata used for defining the order model
  - Purge orders from the database
  - Validate the OSM data model
  - Create a graphical representation of a data model
– Add or remove users from workgroups
– Purge data from the database
– Migrate data

**Note:** Although these functions can be done using the XML Import/Export application, Oracle Communications Design Studio is the preferred application for executing these functions.

- Use the Oracle Scripter to:
  - Run scripts to configure OSM and manage database partitions
  - Create or load objects into the OSM database

### Directory Placeholders Used in This Guide

Table 1–1 shows the placeholders used in this guide to refer to directories related to the OSM application.

<table>
<thead>
<tr>
<th>Placeholder</th>
<th>Directory Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSM_home</td>
<td>The directory into which OSM was installed. This directory contains the SDK directory (if the SDK was installed) and various installation-related files.</td>
</tr>
<tr>
<td>MW_home</td>
<td>The location where the Oracle Middleware product was installed. This directory contains the base directory for the WebLogic Server, a utils directory, and other files and directories.</td>
</tr>
<tr>
<td>WLS_home</td>
<td>The base directory for the WebLogic Server core files. It is located in the MW_home directory.</td>
</tr>
<tr>
<td>domain_home</td>
<td>The directory which contains the configuration for the domain into which OSM is installed. The default location is MW_home/user_projects/domains/domain_name but it is frequently set to some other directory at installation.</td>
</tr>
</tbody>
</table>
This chapter describes how to start and stop the Oracle Communications Order and Service Management (OSM) product.

**About Starting and Stopping OSM**

Because OSM resides on Oracle WebLogic servers, starting or stopping the appropriate WebLogic server also starts or stops OSM.

- **Note:** If your OSM environment is in a WebLogic Server cluster, consult the Oracle WebLogic Server documentation for information about how to start and stop the necessary servers. If all of the servers in the cluster start properly, it means that OSM has started properly.

**Starting the OSM Server**

To start the OSM Server:

1. Change to the `domain_home` directory. This is the base directory for the WebLogic domain into which OSM was deployed.
2. Run the following command:
   ```bash
   nohup ./startWebLogic.sh &
   ```
   This starts the WebLogic server in the background and so that it will not stop if you close the terminal window.

**Verifying that OSM Has Started**

To verify that the OSM application has finished starting, you can use either the following method or the method described in "Using the WebLogic Console to Determine the Status of the OSM Application".

1. From the directory where you started OSM, run the following command:
   ```bash
   tail -f nohup.out
   ```
   Wait until the following text is displayed. It indicates that server startup is complete.
   ```
   <Server started in RUNNING mode>
   ```
   2. Use Ctrl + c to halt the `tail` command and return to the command prompt.
Stopping OSM

To stop OSM:

1. Change to the `domain_home/bin` directory.
2. Run the following command:
   ```
   ./stopWebLogic.sh
   ```

---

**Note:** The procedure above stops OSM by stopping the Administration server for the WebLogic Server. If the WebLogic Server does not shut down completely, you will not be able to start it again due to a port conflict. If the procedure above has completed, but some WebLogic Server processes are still running for the domain, you can use the `kill` command to stop them. See "Verifying that OSM Has Stopped" for information about verifying whether OSM and WebLogic have stopped completely.

---

Verifying that OSM Has Stopped

To verify that OSM has stopped, do any of the following:

- Look at the text in the `nohup.out` file located in the directory where OSM was started (usually `domain_home`). One of the last messages displayed when the server shuts down is:
  ```
  <Order Management Webservice destroyed.>
  ```

- Try connecting to the WebLogic console. If you cannot, WebLogic is probably not running.

- Look at the process list for the user who started the server. If WebLogic is running, there will probably be at least one process with `startWebLogic.sh` in its description.

- Look in the user’s process list for a java process that was started out of the Java directory for WebLogic. Process descriptions vary from platform to platform, so look at the process list when you know OSM is running to see what the entries look like on your platform. You can later use this information to confirm that the WebLogic server has shut down completely.
This chapter describes how to set up security on your Oracle Communications Order and Service Management (OSM) system.

About OSM Security

You use the Oracle WebLogic Server Console to manage OSM security. When you manage OSM security, you can perform the following tasks:

- Add users to groups. See "Adding Users to OSM".
- Set up Secure Sockets Layer (SSL). See "Using Secure Sockets Layer".
- Integrate additional security measures; for example, LDAP servers, Microsoft domain servers, or Linux security.

**Note:** If you use an external security implementation such as LDAP, you should also use a caching realm to improve performance. See "Setting Up a Caching Realm" for more information.


For more information about WebLogic security realms, refer to the WebLogic Server Console documentation.

**Note:** OSM supports LDAP Version 2.

Adding Users to OSM

To add a user to OSM:

- Add users to groups in the WebLogic Server Administration Console.
- Create workgroups as roles in Oracle Communications Design Studio.
- Assign users to workgroups in the OSM Administrator.

Adding Users to Groups in the WebLogic Server Console

All security for OSM users and groups is managed through the WebLogic Server Console. See the Oracle WebLogic Server documentation for more information about creating and deleting users and groups.
To add users to groups:

1. Log in to the WebLogic Server Administration Console.
   You must be a WebLogic administrator.

2. In the Domain Structure tree, click **Security Realms**.
   The Summary of Security Realms page is displayed.

3. Click **myrealm**.
   The settings for the security realm are displayed.

4. Click the **Users and Groups** tab.
   A list of users that have been configured is displayed.

5. Click on a user.
   The user’s description, password, and group membership is displayed. Users are assigned to one or more parent groups that have different levels of access to WebLogic resources, depending on their roles and the tasks they can perform. Groups in the WebLogic security realm represent the roles.

6. On the page that displays the settings for the selected user, click the **Groups** tab.

7. Select a group or groups from the **Available** list, click the right arrow to move the selected group to the **Chosen** list. See "Required Groups and Users" for more information.

8. Click **Save**.

### Creating Workgroups as Roles in Design Studio

In Design Studio, you create roles and assign permissions to give users in that role access to related functions in the Task, and Order Management Web Clients.

---

**Note:** You assign individual users to roles using the Administrator application. See *OSM Administrator Application User’s Guide* for more information.

---

Table 3–1 describes the Web client functions to which you provide access. Most of these are applicable only to the Task Web Client. Only the Reference Number Modification permission is also applicable to the Order Management Web Client.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create Versioned Orders</td>
<td>Enables users to create orders for different versions of cartridges. If not granted this permission, users can only create orders for the default version of the cartridge.</td>
</tr>
<tr>
<td>Exception Processing</td>
<td>Enables users to alter the flow of a process by applying exception statuses at any time throughout the process.</td>
</tr>
<tr>
<td>Online Reports</td>
<td>Enables users to view summarized reports on all orders and tasks on the system.</td>
</tr>
<tr>
<td>Order Priority Modification</td>
<td>Enables users to modify the priority of a task in an order.</td>
</tr>
<tr>
<td>Reference Number Modification</td>
<td>Enables users to modify the reference number of an order.</td>
</tr>
</tbody>
</table>
In addition to granting Web client permissions, you can also grant permissions at the order level (by associating a role to an order type) and the task level. See the discussion about creating new roles in the Design Studio Modeling OSM Processes Help for more information. After you create a role, you must assign permissions to the role entities. See the description of the Role Editor Role tab in the Design Studio Modeling OSM Processes Help for more information about permissions for role entities.

### Assigning Users to Workgroups in OSM Administrator

See the discussion about assigning users to a workgroup in *OSM Administrator Application User’s Guide* for more information.

### Required Groups and Users

When you install OSM, required groups and users are created automatically. To use another security implementation such as LDAP, you must create the appropriate groups and manually add users to them.

Table 3–2 lists the groups and the members of each group.

<table>
<thead>
<tr>
<th>Group Name</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMS_client</td>
<td>All users of the OSM Web Client or OSM HTTP XML API must belong to this group.</td>
</tr>
<tr>
<td>OMS_workgroup_manager</td>
<td>Gives members the ability to create and modify workgroups using the OSM Administrator. They cannot add users to workgroups.</td>
</tr>
<tr>
<td>OMS_user_assigner</td>
<td>Gives members the ability to add users to workgroups from the OSM Administrator.</td>
</tr>
<tr>
<td>OMS_log_manager</td>
<td>Gives access to the log4jAdmin web page for reading log messages.</td>
</tr>
<tr>
<td>OMS_designer</td>
<td>You can do all process, order modeling, and system maintenance. You cannot create or modify workgroups, or assign users to workgroups.</td>
</tr>
</tbody>
</table>
OSM supports three levels of security for interactive users:

- **No use of SSL.** If no SSL support is chosen, the HTTP connection is made on the WebLogic server non-secure port.

- **SSL for all communications.** Complete SSL support is provided by the WebLogic server. An HTTPS connection is made to the SSL port for that server and all communication takes place using SSL on that port.

- **SSL during login only.** Provides a connection to the secure HTTP port. You can use SSL during authentication so user names and passwords are encrypted. After authentication is complete, a non-secure HTTP connection is used. This avoids the performance overhead of using SSL for all communication but gives a level of security for user and password protection. If you choose this option, extra configuration is required.

### Changing Secure Sockets Layer Configuration in OSM

You configure SSL for OSM using the XML parameter file `web.xml` which is stored in the `oms.war` file inside the `oms.ear` file. To edit the `web.xml` file, you must unpack the `oms.ear` file, edit the `web.xml` file, and then repack the `oms.ear` file and re-deploy it to the OSM server.

To change SSL configuration on the OSM server:

1. Locate and extract the `web.xml` file from the `oms.ear/oms.war` file.
2. Add or modify the parameter `secure_login` in the `web.xml` file.
3. Edit the parameter value as follows:
   - For no SSL support, enter 0.
   - For SSL support, enter 1.
4. Save the file.
Setting Up a Caching Realm

If you use an external security implementation such as LDAP, you should also use a caching realm to improve performance. A caching realm holds the results of security checks in memory so that subsequent checks are not required to communicate directly with an external security server. The default settings for caching realms are appropriate for small numbers of users in the external realm; however, they do not help if your external security implementation has large numbers of users.

To set up a caching realm:

1. Log in to the WebLogic Administration Console.
2. In the Domain Structure tree, select Security Realms.
   - The Summary of Security Realms page is displayed.
3. Select the realm from the table.
   - The Settings for myrealm window is displayed.
   - From this window, you can change the settings for your realm.

Using the XML Import/Export Application to Administer Users and Workgroups

The userAdmin command lets you add users to WebLogic groups and OSM workgroups using an XML document. The XML document contains the user information you want to add or configure based on the UserAdmin.xsd schema, which can be found in the \SDK\XMLImportExport\models directory.

Administering users in this way allows you to manage users in volume instead of assigning them individually, and it permits the integration of OSM users into a larger, enterprise system administration application.

You must encrypt the passwords in the source XML document prior to running the userAdmin command. Otherwise, when you run userAdmin, you will trigger an error indicating that the EncryptPassword utility should be run. See "Using the EncryptPasswords Utility" for more information.

To run the userAdmin command, you should first modify the classpath in the config.bat file (if you are using the command-line script) or the xml.project.class.path in the build.xml file (if you are using Ant), to ensure the following:

- Contains only one reference to the weblogic.jar file associated with the WebLogic installation to which the command is connecting. If there is no such reference, add it. If there is any reference pointing to the one under XMLImportExport/lib, change it.
- Contains only one reference to the ojdbc<X>.jar file associated with the WebLogic instance to which the command is connecting. If there is one pointing to the one under XMLImportExport/lib, change it.
- Contains a reference to XMLImportExport/classes/schemaTypes.jar. If there is no such reference, add it.
- Remove any references to XMLImportExport/lib/nls_charset12.jar and XMLImportExport/lib/classes12.zip.

To set up your environment and run the command:
1. Define an XML file with the user information to add, similar to the sample shown below. See UserAdmin.xsd for more details on the XML schema.

```
<userConfig xmlns="http://www.metasolv.com/Provisioning/UserConfig"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.metasolv.com/Provisioning/UserConfig
C:\Provisioning\SDK\XMLImportExport\models\UserAdmin.xsd">
  <user name="demo">
    <description>A test user</description>
  </user>
  <clientGroup>
    <user>demo</user>
  </clientGroup>
  <designerGroup>
    <user>demo</user>
  </designerGroup>
  <logManagerGroup>
    <user>demo</user>
  </logManagerGroup>
  <userAssignerGroup>
    <user>demo</user>
  </userAssignerGroup>
  <workgroupManagerGroup>
    <user>demo</user>
  </workgroupManagerGroup>
  <workgroup name="demo">
    <user>demo</user>
  </workgroup>
  <workgroup name="everyone">
    <user>demo</user>
  </workgroup>
</userConfig>
```

2. Encrypt the passwords in your source XML file. See "Using the EncryptPasswords Utility".

3. If required, edit the config.xml file to modify your connection information to WebLogic. You must have the authority to administer users in WebLogic.

```
<j2eeAdminConnection>
  <user>weblogic</user>
  <password>password</password>
  <hostname>localhost</hostname>
  <port>7001</port>
</j2eeAdminConnection>
```

**Note:** The <password> element in the config.xml file is optional. If you opt not to specify the password in the <password> element, you can specify it in a command prompt as follows:

```
-j2eeAdminPassword <password>
```

4. Run the following command at the system prompt:

```
userAdmin [XML_MODEL] [XML_CONFIG]
```

Where:
Using the EncryptPasswords Utility

You use the EncryptPasswords utility to encrypt the user name and password credentials of XML Import/Export application users.

For information on the EncryptPasswords utility, see the following:

- About the EncryptPasswords Utility
- Running the EncryptPasswords Utility

About the EncryptPasswords Utility

The EncryptPasswords utility (located in the OSM_home\SDK\XMLImportExport directory) is a password management utility that secures the credentials that the XML Import/Export application uses to access the OSM database, the XML API interface, and the WebLogic domain Administration Server. The EncryptPasswords utility encrypts the credentials, preventing their accidental exposure.

Running the EncryptPasswords utility script stores the user names and passwords of all XML Import/Export application users in encrypted format in the configuration file which the XML Import/Export application uses to provide these credentials (for example, OSM_home\SDK\XMLImportExport\config\config.xml). When the XML Import/Export application runs, it decrypts the passwords as part loading its configuration file.

The EncryptPasswords utility can be run only by a user who has write access to the XML files in which the credentials are stored.

When you install the XML Import/Export application, you can optionally provide passwords for the OSM database, the XML API interface, and the WebLogic domain Administration Server. To set and reset those passwords, you run the EncryptPasswords utility. See "Running the EncryptPasswords Utility" for information on running the utility.

The EncryptPasswords utility prompts you to enter the user name and password credentials of each XML Import/Export application user that requires access to the OSM database, the XML API interface, and the WebLogic domain Administration Server. The utility then encrypts the credentials and stores their encrypted form in the configuration file specified when the utility is run.

Ant build files for the EncryptPasswords utility are located in the following directories:

- XML_MODEL is the name of the XML model document containing the user information.
- XML_CONFIG is the name of the configuration file.

For example:

userAdmin data/users.xml config/config.xml

or

ant userAdmin

5. In the OSM Administrator, use the refresh buttons to refresh the Users and Workgroups lists, or use ant Refresh in the CDK to refresh the metadata.

6. After the users have been created, you can assign functions and tasks to them using Design Studio.
Using the EncryptPasswords Utility

- $OSM_home\SDK\CartridgeManagement\production$
- $OSM_home\SDK\CartridgeManagement\development$

The Ant build files have targets corresponding to each of the batch files in the $OSM_home\SDK\XMLImportExport$ directory that include the EncryptPasswords functionality.

Running the EncryptPasswords Utility

Run the EncryptPasswords utility script:

- As part of the initial setup of the XML Import/Export application
- Each time the user name or password credentials of an XML Import/Export application user changes

See "About the EncryptPasswords Utility" for information on how the EncryptPasswords utility works.

---

**Note:** To run the EncryptPasswords utility, you have write access to the XML files in which the XML Import/Export application user credentials are stored.

---

To run the EncryptPasswords utility, use these arguments and syntax:

```
EncryptPasswords [XML_CONFIG] OPTIONAL{-dbUser} OPTIONAL{-xmlapiUser} OPTIONAL{-wlsUser}
```

where the XML Import/Export application password is set and reset for gaining access to

- **-dbUser:** the OSM database
- **-xmlapiUser:** the XML API interface
- **-wlsUser:** the WebLogic domain Administration Server

and [XML_CONFIG] is the example XML Import/Export application configuration XML file (config_sample.xml) that you copied to a new file and renamed (for example, config.xml).

When you set a user’s credentials, you specify only the systems that they use for the XML Import/Export application operations they perform. For example, if the user only imports or exports cartridges, you only need to specify the **-dbUser** flag.

Removing an Encrypted Password

To remove a user name and password for a user that no longer requires credentials, open the XML file where the credentials are stored and remove them manually. If you do not remove them manually, the user name and password combination continues to exist in the XML file.
Securing Credentials Required to Access External Systems

Oracle Communications Order and Service Management (OSM) applications, such as OSM Web clients and OSM cartridge applications, often are required to provide credential information to gain access and log in to external systems. The credential information must be secure and cannot be hard coded in OSM code. This chapter describes how to secure credentials for accessing external systems by using a credential store, through the Fusion Middleware Credential Store Framework (CSF).

About the Credential Store

A credential store is a central repository you can use to store credentials such as user names and passwords. OSM can use a credential store for the credentials needed to log in to and send requests to external systems (meaning any resources or systems that are out of the scope of your OSM cartridge-owned code and resources). These credentials can be used by OSM and OSM applications.

The type of credential store that OSM uses is offered through the Oracle Platform Security Services (PSS) and is part of the CSF security service available to WebLogic Server. CSF offers both file-based and LDAP-based credential stores. OSM uses the file-based credential store by default. Oracle recommends using LDAP-based stores in a production environment.

The CSF is a set of APIs that applications can use to create, read, update, and manage credentials securely. OSM includes wrapper APIs to the CSF APIs. Use the OSM credential store APIs when developing your OSM cartridges so your OSM applications can access and read credentials from the credential store in a secure form. For example, use the OSM credential store APIs when you define data provider classes in your cartridges which must access Web Service and database systems with credentials. See "Developing Cartridges to Use the Credential Store" and "OSM Credential Store Command and API Reference Material" for more information on using the credential store in your cartridge development.

Oracle recommends you to use the credential store as a repository for credentials and use the OSM credential store APIs in OSM-related code to access the repository in a secure form. Oracle strongly recommends you do not hard code user names and passwords in cartridge code and that you update any current cartridges that have hard coded credentials to use the OSM credential store APIs. See "Developing Cartridges to Use the Credential Store" for more information on security options and recommendations.

See the Oracle Fusion Middleware Application Security Guide at:

http://download.oracle.com/docs/cd/E12839_01/core.1111/e10043.pdf


for information on PSS and managing credentials in the credential store.

**How OSM Retrieves Credentials from the Credential Store**

The credential store in the WebLogic domain for OSM applications contains credentials which are stored using a credential store map and key names. OSM applications, such as OSM Web clients and OSM cartridge applications, retrieve credentials from the credential store based on the credential store map and key names. Automation plug-ins are used to call OSM credential store APIs to retrieve the credentials. OSM applications use the credentials to gain access to external systems.

OSM credential store APIs are used inside automation plug-ins to retrieve credentials and to gain access to external systems. OSM cartridge code can call the credStoreAdmin command (an XML Import/Export application command) to create and configure the required entries, such as map name, user name, and password, for OSM applications in the credential store. See "credStoreAdmin Command" for information on this utility.

OSM plug-in users in a cartridge application, such as automation plug-ins or external instance adapters access the OSM credential store map and read the credentials.

The following steps summarize how an OSM automation plug-in retrieves credentials from the credential store for an automation task that requires the credentials to access an external system:

1. The automation plug-in script uses the getCredential or getOsmCredentialPassword method of the AutomationContext API.
2. WebLogic Server checks the JPS (Java Platform Security) policy and confirms the automation plug-in has access permissions from the OSM domain to the credential store.
3. The automation plug-in user accesses the correct credential store map and reads the credentials required to access the external system.
   
   If the credentials are not in the store, the API fails with an exception and the automation task fails.
4. If the credentials are in the store, the user name and password variables in the plug-in script will be set with values retrieved from the credential store.
5. The message is sent to the external system with the credential information.
6. The automation task completes.

**Using the Credential Store in OSM**

OSM uses the credential store offered through CSF (see "About the Credential Store"). You can use the credential store to store the user name and password information that OSM and its applications require to log in to and send a request to external systems.

See "About the Credential Store" for a description of the credential store OSM uses by default.

See "How OSM Retrieves Credentials from the Credential Store" for information on how OSM retrieves the credentials from the credential store.

See "Managing Credentials in the Credential Store" for information on adding credentials to the credential store and managing credentials.
See "Developing Cartridges to Use the Credential Store" for information on developing OSM cartridges to use the credential store.

See "Upgrading Existing Cartridge Code to Use the Credential Store" to update your existing cartridge code to use the credential store.

**Configuring the Java Security Policy for the OSM Credential Store**

For OSM and its plug-in applications to read data from the credential store, the WebLogic server must be configured to grant them Java permissions to access the credential store map. In PSS, credential store access permission is configured through the JPS policy. The default JPS policy instance is in the $domain.home/config/fmwconfig/system-jazn-data.xml file.

When you configure the JPS policy, you specify the code source directory of the OSM application (oms); this grants credential store access permission to all JAR files in the code source directory.

The default credential store map name for OSM applications is osm. For OSM plug-in users to access the default map, configure the JPS policy using map name (osm). You can create your own credential store map in the credential store. For OSM plug-in users in cartridge applications to access your credential store map and read the credentials, configure the JPS policy using your own map name.

In a clustered environment, use the administration server URL and not the managed server when you configure the credential store and the JPS policy.

To configure the JPS policy, you can use either of the following methods:

- The credStoreAdmin command. See "Configuring the Java Security Policy Using credStoreAdmin Command" to configure the JPS policy for the OSM credential store map using this command. See "credStoreAdmin Command" for a description of this command.

- The Fusion Middleware WebLogic Scripting Tool grantPermission command. See the Oracle Fusion Middleware Security Guide for information on managing policies with the WebLogic Scripting Tool commands.

- The Fusion Middleware Control GUI. Credential store and JPS policy MBeans can be configured from this GUI. See the Oracle Fusion Middleware Application Security Guide at:

  http://download.oracle.com/docs/cd/E12839_01/core.1111/e10043.pdf

  for information.

**Verifying the Java Security Policy Configuration**

To verify the JPS policy is configured for your credential store:

Open the $domain.home/config/fmwconfig/system-jazn-data.xml file and verify that it contains the following lines in bold text:

```
<grant>
  <grantee>
    <codesource>
      Note: This JPS policy entry shows credential store access from the OSM application (oms) to the default OSM credential store map osm in the credential store.
    </codesource>
  </grantee>
</grant>
```
Do not update the JPS policy configuration manually by editing system-jazn-data.xml directly. Use the methods described in "Configuring the Java Security Policy for the OSM Credential Store" to configure the JPS policy.

### Configuring the Java Security Policy Using credStoreAdmin Command

To configure the JPS policy for the OSM credential store map using the credStoreAdmin command batch script (recommended method):

**Note:** This procedure assumes you have:

- Installed OSM.
- Installed OSM SDK.

1. Create and edit the configuration file for credStoreAdmin command (config.xml). See "Creating the Configuration File for the credStoreAdmin Command".

2. Create and edit the XML data file for credStoreAdmin command (credential.xml). See "Creating the XML Data File for the credStoreAdmin Command". The JPS policy and credentials are configured based on the data in the credential.xml file.

3. From the XML Import/Export application directory (OSM_home/SDK/XMLImportExport), set the following required environment variables for the credStoreAdmin command in config.bat (Windows) or config.sh (Unix):
   - JAVA_HOME
   - APP_ROOT: Point to the XMLImportExport directory in the OSM SDK installation.
   - MIDDLEWARE_HOME: Point to the Fusion Middleware installation directory.

4. Set the required properties in the build.properties file. See "Setting Build Properties for Credential Store Commands".

5. Run the credStoreAdmin batch script as follows:
   ```
   credStoreAdmin credential.xml config/config.xml
   ```

6. When prompted, enter the password for WebLogic user.

7. Verify the JPS policy is configured for your credential store as described in "Verifying the Java Security Policy Configuration".


You can also configure the JPS policy for the OSM credential store map using the credStoreAdmin command Ant script.
To configure the JPS policy for the OSM credential store map using the `credStoreAdmin` command Ant script:

---

**Note:** This procedure assumes you have:
- Installed OSM.
- Installed OSM SDK.

---

1. Create and edit the configuration file for the `credStoreAdmin` command (`config.xml`). See "Creating the Configuration File for the `credStoreAdmin` Command".

2. Create and edit the XML data file for the `credStoreAdmin` command (`credential.xml`). See "Creating the XML Data File for the `credStoreAdmin` Command".

3. From the XML Import/Export application directory (`OSM_home/SDK/XMLImportExport`), set required properties in the `build.properties` file as described in "Setting Build Properties for Credential Store Commands".

4. Run the `credStoreAdmin` Ant script and provide the WebLogic administrator user password in the command line as follows:

   ```bash
   > ant -Dwls_admin_password=password credStoreAdmin
   ```

5. Verify the JPS policy is configured for your credential store as described in "Verifying the Java Security Policy Configuration".


You can call the `credStoreAdmin` target in another Ant script. For an example of calling the `credStoreAdmin` Ant script in your own Ant script, see "Calling `credStoreAdmin` Target in Another Ant Script".

---

**Managing Credentials in the Credential Store**

You must provision the credential store with credentials before OSM applications can use them. Provision the credential store by adding the user names and passwords required for OSM and external systems. You must also update passwords in the credential store every time you change user passwords through the WebLogic Server console.

Store all OSM users in the default OSM map, `osm`, and use a default value in key name `osmUser_+username`.

To manage credentials, such as adding, updating, or deleting credentials for OSM and external systems in the credential store, do either of the following:

- Use Oracle WebLogic Scripting Tool commands provided by WebLogic Server. See [Oracle Fusion Middleware Security Guide](http://download.oracle.com/docs/cd/E12839_01/core.1111/e10043.pdf) at:

  ```
  http://download.oracle.com/docs/cd/E12839_01/core.1111/e10043.pdf
  ```

  for information on managing credentials in the credential store by using WebLogic Scripting Tool commands.

- Use the `credStoreAdmin` command in the XML Import/Export application. See "`credStoreAdmin Command`" for information on this command.

To manage credentials in the credential store using `credStoreAdmin` command:
1. From the XML Import/Export application directory (OSM_home/SDK/XMLImportExport), set required properties in the build.properties file as described in "Setting Build Properties for Credential Store Commands".

2. From the XML Import/Export application directory (OSM_home/SDK/XMLImportExport), run the script credStoreAdmin:

```
> credStoreAdmin credential.xml config/config.xml
```

where credential.xml is the XML data file that includes the map, key, user name, and password values. See "Creating the XML Data File for the credStoreAdmin Command".

The following is an example of the XML data file:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns2:CredentialConfig
 xmlns:ns2="http://www.metasolv.com/Provisioning/CredentialConfig"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation=".../XMLImportExport/models/CredStoreAdmin.xsd">
 <ns2:credential operation="create">
  <ns2:mapname>osm</ns2:mapname>
  <ns2:keyname>osmUser_osm</ns2:keyname>
  <ns2:user>osm</ns2:user>
  <ns2:password/>
 </ns2:credential>
 <ns2:credential operation="create">
  <ns2:mapname>osmlf</ns2:mapname>
  <ns2:keyname>osmUser_osmlf</ns2:keyname>
  <ns2:user>osmlf</ns2:user>
  <ns2:password/>
 </ns2:credential>
</ns2:CredentialConfig>
```

where config.xml is the configuration file that contains the data for WebLogic Server. See "Creating the Configuration File for the credStoreAdmin Command".

The following is an example of the config.xml file:

```xml
<j2eeAdminConnection>
  <user>weblogic</user>
  <password/>
  <hostname=localhost</hostname>
  <port>7001</port>
</j2eeAdminConnection>
```

To manage credentials in the credential store using the userAdmin command:

1. From the XML Import/Export application directory (OSM_home/SDK/XMLImportExport), set required properties in the build.properties file as described in "Setting Build Properties for Credential Store Commands".

2. Create and edit the configuration file for the userAdmin command (config.xml). See "Creating the Configuration File for the userAdmin Command".
3. Create and edit the XML data file for the userAdmin command (user.xml). See "Creating the XML Data File for the userAdmin Command".

It is recommended that you do not enter password values in user.xml. By leaving password values empty, you can enter passwords interactively at run time.

4. From the XML Import/Export application directory (OSM_home/SDK/XMLImportExport), do either of the following:
   - (Recommended method) Run the batch script userAdmin and input password information when prompted:
     
     ```
     >userAdmin user.xml config/config.xml
     ```
   - Run the Ant script userAdmin and provide password information in the command line:
     
     ```
     >ant -Dwls.admin.password=password -Dxmlie.root.dbPassword=password userAdmin
     ```

If the Ant target userAdmin is called by an Ant script that is running in your Oracle Communications Design Studio workspace, interactive mode is not supported and passwords must be entered in the XML data file. In this case, delete the XML data file immediately after use. For an example of how to call the userAdmin command Ant script in another Ant script, see "Calling UserAdmin Target in Another Ant Script".

### Developing Cartridges to Use the Credential Store

When your OSM cartridges require data from external systems, trigger actions at external systems, or provide data to external systems, credential information may be required by the external system. The external system can be any resource or system that is out of the scope of your OSM cartridge-owned codes and resources.

When you develop OSM cartridges, Oracle recommends you use the credential store to allow plug-in code to access credential information in a secured way. You can use the OSM credential store APIs for code that requires credential retrieval.

OSM uses the credential store offered through WebLogic Server; however, you are not required to use this credential store to secure credentials. You can use other methods of securing credentials. Oracle strongly recommends you do not hard code user credential information in OSM code such as in plug-in script files and cartridge model description files. Passing and storing passwords in plain text poses a security risk. Follow proper security guidelines to develop OSM cartridges to protect data over communication channels. Oracle recommends using SSL communication between OSM and an external system, particularly for Web Services of external systems.

The following are examples of external systems used in OSM cartridges that may require credential information:

- OSM Web Service
- Databases
- JMS queues and topics (excepts JMS queues deployed by the cartridge)
- Web Services of any system

To develop your OSM cartridges to use the credential store, see the following:
- Use "AutomationContext" in your automation plug-in code to retrieve credentials from the credential store. See Developing Automation Plug-ins to Use the Credential Store for more information.

- Use the operation APIs in "ViewRuleContext" in XQuery scripts to access credentials stored in the credential store.

- Use "PasswordCredStore" in your JAVA classes to retrieve user name and password from the credential store.

- Use the attributes for credential store in "SoapAdapter" to retrieve credentials from the credential store when sending a SOAP request using HTTP/HTTPS.

- Use the attributes for credential store in "ObjectelHPPTAdapter" to retrieve credentials from the credential store when sending a request to Objectel. See "Defining Data Providers in OSM Cartridges to Use the Credential Store" for more information.

- See "OSM Credential Store Command and API Reference Material" for a description of the OSM credential store APIs.

See "About the Credential Store" for information about the credential store.

Developing Automation Plug-ins to Use the Credential Store

Some OSM credential store APIs can be used in custom automation plug-in Java classes. Use these APIs when you define custom automation plug-in classes to access an external system with credentials. You can also call OSM credential store APIs from your automation context classes. The XQuery plug-in code for an automation or activation task can use credential store APIs to retrieve credentials from the credential store. See "AutomationContext" for example code of developing automation plug-ins in OSM cartridges to retrieve credentials from the credential store.

External instance adapters and automation plug-in classes (XQuerySender and XSLTSender) provided by Oracle to send messages and requests to external systems support the OSM credential store APIs.

Defining Data Providers in OSM Cartridges to Use the Credential Store

A data provider class in your cartridge where credential information is required by an external system must be set up to read the required credentials from the credential store. For information on setting up your data provider classes to be able to read from the credential store, see the following:

- Using the Credential Store with Custom Data Providers

- Using the Credential Store with Built-In Data Providers

Using the Credential Store with Custom Data Providers

When you add a data provider class in your cartridge where credential information is required by an external system, the data provider class can call OSM credential store APIs to read the required credentials from the credential store. Your data provider class must implement method retrieveInstance() of interface ExternalInstanceAdapter.

To read the required credential from the credential store when you define your own data provider class (provider type is "Custom"), use the following APIs in the retrieveInstance() method in your Java class:
public Element retrieveInstance(final ViewRuleContext context, final Map<String, Object> parameters) throws Exception {
    // DoCustomLogic

    String mapname = getStringParam(parameters, "para.mapname", null);
    String keyname = getStringParam(parameters, "para.keyname", null);
    String username = "";
    String password = "";

    if (mapname != null && keyname != null) {
        try {
            String credential = context.getCredential(mapname, keyname);
            int index = credential.indexOf("/");
            username = credential.substring(0, index-1);
            password = credential.substring(index);
        } catch (Exception e) {
            // DoCredStoreExceptionHandling
        }
    }

    // DoAuthenticationWithUsernamePassword
    // DoCustomerRequest
    // DoResponseHandling
}

Using the Credential Store with Built-In Data Providers

Oracle provides pre-defined or built-in data provider classes "SoapAdapter" and "ObjectelHPPTAdapter" which contain the code required for using the credential store. To use the credential store when you use these built-in adapters, add the input parameters required for the credential store in your data provider.

If you use the SoapAdapter, add the input parameters:

- Name of parameter "oms:credentials.mapname", contentType is "XPATH", default value is "myMap" (this example uses a custom map)
- Name of parameter "oms:credentials.keyname", contentType is "XPATH", default value is "myUser"

If you use the ObjectHTTPAdapter, add the input parameters:

- Name of parameter "obj:mapname", contentType is "XPATH", default value is "osm" (this example uses the default map)
- Name of parameter "obj:mapname", contentType is "XPATH", default value is "osmUser_osm" (this example uses user "osm" stored in default map)

Upgrading Existing Cartridge Code to Use the Credential Store

Upgrade your existing OSM cartridge code to use the credential store by using the OSM credential store APIs. Upgrade your custom data provider classes and XQuery plug-in code to use the OSM credential store APIs for retrieval of credentials. See

Note: This example assumes you are using your own map. If you use the default map (osm) and key names for the OSM application, you can use simpler code:

String password = context.getOsmCredentialPassword(username)
"Developing Cartridges to Use the Credential Store" for information on developing cartridges to use the credential store.

In addition to upgrading your cartridge code, provision the credential store for the WebLogic domain for OSM applications with required credentials (see "Managing Credentials in the Credential Store") and configure the JPS policy for the WebLogic domain to allow OSM access to the credential store (see "Configuring the Java Security Policy for the OSM Credential Store").

Using Built-in SOAP Adapter as a Data Provider Class

Credential information is required to send a SOAP request. If your existing automation plug-in code that is used to test the SOAP adapter has hard coded passwords, you can use built-in SOAPAdapter as a data provider class in your cartridges to remove the need for the hard coded passwords.

When you use the default map for OSM applications, automation plug-in users pass in the user name only in the parameter. When you use your custom credential store map, automation plug-in users pass in the credential map name and key name for credentials in the parameter.

To update existing automation plug-in code that tests SOAPAdapter to remove hard coded passwords:

1. Remove hard coded passwords from the existing "oms:credentials:password" input parameter.

2. Ensure that credentials exist in the credential store under map="osm" key="osmUser_"+<SoapRequest_username>.

3. Test that the SOAP adapter works correctly after the update.

OSM Credential Store Commands

OSM offers two credential store commands available in the XML Import/Export application which can be used for OSM credential store setup or management:

- **credStoreAdmin command**
  
  You can use this command to configure the JPS policy for the credential store and manage credentials in the credential store. See "credStoreAdmin Command" for a description of this command.

- **userAdmin command**
  
  You can use this command for credential store management. This command is also used to administer OSM users and workgroups. See "userAdmin Command" for a detailed description of this command.

See "Using the XML Import/Export Application" for information on using the XML Import/Export application.

Setting Build Properties for Credential Store Commands

To use the credential store commands credStoreAdmin and userAdmin, you must set the following properties in OSM_home/SDK/XMLImportExport/build.properties:

- **xmlie.root.dir**: The XML Import/Export application directory (/XMLImportExport) in the OSM SDK installation.

- **middlewareHome**: The Fusion Middleware installation directory.
- **xmlie.root.modelDocument**: The XML data file that contains the credential information (*credential.xml*).

- **xmlie.root.configDocument**: The configuration file for the credStoreAdmin command (*config/config.xml*).

Note that the credStoreAdmin command does not use the other properties in the *build.properties* file.
To develop OSM cartridges to use the credential store offered through CSF (see “About the Credential Store”), use the OSM credential store APIs. OSM credential store APIs are wrapper APIs to the CSF APIs. Use the OSM credential store APIs in your OSM-related code that requires credential retrieval, such as in data providers and automation plug-ins.

The OSM credential store APIs and credential store-related classes are:

- userAdmin Command
- credStoreAdmin Command
- CredStore
- PasswordCredStore
- CredStoreException
- SoapAdapter
- ObjectelHPPTAdapter
- ViewRuleContext
- AutomationContext
**userAdmin Command**

The userAdmin command is part of the XML Import/Export application and is used to administer OSM users and workgroups. The userAdmin command also supports credential store management.

Use the userAdmin command to create an OSM user and also add the user in the credential store.

For credential-store related interface and object details, see "J2ee Manager/WLUserManager" and "UserAdminOperation".

Use the userAdmin command to add OSM users to the default OSM credential store (to the default map with default key values).

**Syntax**

Batch script:

```
userAdmin user.xml config/config.xml
```

Ant script:

```
ant userAdmin
```

**Creating the XML Data File for the userAdmin Command**

To create the XML data file for the userAdmin command (**user.xml**):

```
<userConfig xmlns="http://www.metasolv.com/Provisioning/UserConfig"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <user name="testOsmUser1">
    <description>OSM test user 1</description>
    <password>mypassword1</password>
  </user>
  <user name="testOsmUser2">
    <description>OSM test user 2</description>
    <password>mypassword2</password>
  </user>
  <clientGroup>
    <user>testOsmUser1</user>
    <user>testOsmUser2</user>
  </clientGroup>
  <automationGroup>
    <user>testOsmUser1</user>
    <user>testOsmUser2</user>
  </automationGroup>
  <wsAPIGroup>
    <user>testOsmUser1</user>
    <user>testOsmUser2</user>
  </wsAPIGroup>
  <xmlAPIGroup>
    <user>testOsmUser1</user>
  </xmlAPIGroup>
</userConfig>
```

*Important:* Remove the file when the operation is complete because it contains unencrypted passwords.
Creating the Configuration File for the userAdmin Command

To create the configuration file for the userAdmin command (config.xml):

1. Copy the sample XML Import/Export application configuration file config/config_sample.xml and rename it to config/config.xml.

2. Edit the "j2eeAdminConnection" and "log" sections of the file with your installation information.

   The following is an example "j2eeAdminConnection" section which contains the data for WebLogic Server:

   ```xml
   <j2eeAdminConnection>
     <user>weblogic</user>
     <password/>
     <hostname>localhost</hostname>
     <port>7001</port>
   </j2eeAdminConnection>
   ```

   When the password value is empty in the configuration file, which is the recommended approach for security purposes, you must:
   - Input the password at run time when prompted if running the batch script or Ant task.

3. Edit the "credentialStore" section to define the credentialStore element as true:

   ```xml
   <credentialStore addUser="true"/>
   ```

   This enables the userAdmin command to perform credential store updates.

4. (Optional) Edit the "databaseConnection" section.

   If you configure workgroups using the userAdmin command and the XML data file contains "workgroup" sections, you are required to edit this section. However, it is better to avoid configuring workgroups using the userAdmin command because it requires setting up database connection parameters in the configuration file which is not a secure approach. Instead, it is recommended that you configure workgroups after OSM user is created using OSM Administrator or during cartridge deployment.

Usage Notes

The userAdmin command can create a new WebLogic user and add the user to the OSM default credential store map at the same time.

Calling UserAdmin Target in Another Ant Script

The following is an example on how to invoke the userAdmin Ant script in your own Ant script:
<target description="Configure OSM user" name="setupUsers" depends="wls_password">
  <echo message="Create users in WebLogic and Credential Store"/>
  <ant inheritRefs="true" antfile="${xmlieRoot}/build.xml" dir="${xmlieRoot}" target="userAdmin">
    <property name="wls_admin_user" value="weblogic"/>
    <property name="wls_admin_password" value="${wls.password}"/>
    <property name="wls_host" value="localhost"/>
    <property name="wls_port" value="7001"/>
    <property name="middlewareHome" value="${middleware.home}"/>
    <property name="xmlie.root.modelDocument" value="user.xml"/>
    <property name="xmlie.root.configDocument" value="config.xml"/>
  </ant>
</target>

<target name="wls_password">
  <input message="Enter WebLogic Admin User Password: ">
    <handler classname="org.apache.tools.ant.input.SecureInputHandler"/>
  </input>
</target>

---

**J2ee Manager/WLUserManager**

- **Business Object Name:** J2eeManager/WLUserManager
- **Business Object Component Name:** Package name: com.mslv.oms.j2ee.useradmin
- **Description:** This class is used to create J2EE user in WebLogic and add the user to appropriate J2EE groups. It can also add the user in the WebLogic CSF credential store.

### Attributes

- **credStoreName**
  - **Type:** ObjectName
  - **Description:** MBean object for credential store: JpsJmxConstants.MBEAN_JPS_CREDENTIAL_STORE

### Business Object Operations

- **Operation Name:** createUserInCredentialStore
- **Description:** Method which adds the user in credential store.
  
  If the map/key pair exists in the credential store already, it will be overwritten with new values.

**UserAdminOperation**

- **Business Object Name:** UserAdminOperation
- **Business Object Component Name:** Package name: com.mslv.oms.metadatahandler.operation

- **Description:** This class is used to create J2EE user in WebLogic, and add the user to appropriate J2EE groups. It also can add the user in the credential store.

### Attributes

- **OSM_CREDENTIAL_MAPNAME**
Type: String (static final)
Sensitive: Value is "osm"
Description: Pre-defined map name for OSM application in credential store.

- **OSM_CREDENTIAL_KEYNAME_PREFIX**
  Type: String (static final)
  Sensitive: Value is "osmUser_"
  Description: Prefix of key names used for OSM users in credential store.

**Business Object Operations**

Operation Name: configureJ2eeUsers

Description: This method can add users to the credential store.

After a user is created in the J2EE server, a check is made if configuration is defined to add the user in the credential store. The following line is the example configuration (the default value of this configuration is set to "false"):

```xml
<credentialStore addUser="true"/>
```

The user is added to the credential store using the default map name **OSM_CREDENTIAL_MAPNAME** and default key name **OSM_CREDENTIAL_KEYNAME_PREFIX+OSM_username**. For example, if OSM user name is "osmlf", then the map and key values used for it will be map="osm" and key="osmUser_osmlf".
credStoreAdmin Command

Use the credStoreAdmin command to configure the JPS policy for the credential store and to manage credentials in the credential store.

Cartridges can use the credStoreAdmin command to create and configure credential stores during setup.

The credStoreAdmin command is available as an Ant script and as a batch script in the XML Import/Export application (which is included in the OSM SDK package). The batch script supports interactive mode which allows users to input passwords at run time; this is the recommended method of using the credStoreAdmin command because entering the password at run time is a more secure approach.

See "Configuring the Java Security Policy for the OSM Credential Store" for instructions on configuring the JPS policy for the OSM credential store map using the credStoreAdmin command.

See "Managing Credentials in the Credential Store" for information on using the credStoreAdmin command to manage credentials in the credential store.

ANT Task Name
credStoreAdmin

Batch Script Name
credStoreAdmin.bat (Windows)

Schema File
The schema file for the credStoreAdmin command is OSM_home/SDK/XMLImportExport/models/CredStoreAdmin.xsd

Task Arguments
XML data file that contains credential information: credentials.xml (see "Creating the XML Data File for the credStoreAdmin Command" for information on creating this file).
XMILE configuration file: config.xml
WLS administrator password (if not provided in config.xml)
If the WLS administrator password is provided in the command line, the following values can be passed in also:

Note: This mode is used when a cartridge uses this command to create and configure credential stores during setup.

- WebLogic administrator user name
- WebLogic server host
- WebLogic server port
Schema File Input Data Format

The following is the schema for the XML Import/Export application configuration file (config.xsd):

```xml
<xsd:element name="configuration">
  <xsd:complexType>
    <xsd:sequence>
      ...
      <xsd:element name="credentialStore" type="oms:credentialStoreType" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>It determines if user should be added in OSM credential store for new OSM user. The default would be no if node not exist.</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
```

Creating the XML Data File for the credStoreAdmin Command

The following is an example XML data file for the credStoreAdmin command (credential.xml). This example uses the map name `osm`, the default map for OSM applications. If you do not use the default map, replace `osm` with your map name.

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <ns2:jpsPolicy operation="add">
    <ns2:mapname>osm</ns2:mapname>
  </ns2:jpsPolicy>
</ns2:CredentialConfig>
```

Creating the Configuration File for the credStoreAdmin Command

To create the configuration file (config/config.xml):

1. Copy the sample configuration file config/config_sample.xml and rename it to config/config.xml.
2. Edit the "j2eeAdminConnection" and "log" section of the file with your installation information.

   Note that other sections of the file are not used in the credStoreAdmin command, but they must exist and can use dummy values.

   The following is an example "j2eeAdminConnection" section which contains the data for WebLogic Server; for example.
When the password value is empty in the configuration file (which is the recommended approach for better security):

- You must input the password at run time when prompted if running the batch script or Ant task.

**Calling credStoreAdmin Target in Another Ant Script**

The credStoreAdmin target can be called directly in Ant scripts or batch scripts; this capability can be used during an OSM installation with OSM cartridges and custom cartridges.

The following is an example of how to call the credStoreAdmin target in another Ant script:

```xml
<target name="setupJPSPolicy" description="Configure JPS Policy">
  <echo message="Configure JPS Policy for default credential store in WebLogic"/>
  <ant inheritRefs="true" antfile="${xmlieRoot}/build.xml" dir="${xmlieRoot}" target="credStoreAdmin">
    <property name="wls_admin_user" value="weblogic"/>
    <property name="wls_admin_password" value="${wls.password}"/>
    <property name="wls_host" value="localhost"/>
    <property name="wls_port" value="7001"/>
    <property name="middlewareHome" value="${middleware.home}"/>
    <property name="xmlie.root.modelDocument" value="credential.xml"/>
    <property name="xmlie.root.configDocument" value="config.xml"/>
  </ant>
</target>
```

If the Ant target "credStoreAdmin" is called by another Ant script, which is running in your Design Studio workspace, interactive mode is not supported. In this case, passwords must be provided in the XML data file.

Note: It is recommended that you delete this data file immediately after use because it contains unencrypted passwords.

The following is an example of an XML data file that contains the passwords for user osm and osmlf:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns2:CredentialConfig
  xmlns:ns2="http://www.metasolv.com/Provisioning/CredentialConfig"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=" ../XMLImportExport/models/CredStoreAdmin.xsd">
```

---

Securing Credentials Required to Access External Systems 4-19
<ns2:credential operation="create">
  <ns2:mapname>osm</ns2:mapname>
  <ns2:keyname>osmUser_osm</ns2:keyname>
  <ns2:user>osm</ns2:user>
  <ns2:password>osmAdmin</ns2:password>
</ns2:credential>
<ns2:credential operation="create">
  <ns2:mapname>osmlf</ns2:mapname>
  <ns2:keyname>osmUser_osmlf</ns2:keyname>
  <ns2:user>osmlf</ns2:user>
  <ns2:password>osmlfAdmin</ns2:password>
</ns2:credential>
</ns2:CredentialConfig>

Business Object Operations for credStoreAdminOperation
For business operations for credStoreAdminOperation, see "CredStoreAdminOperation".

CredStoreAdminOperation
Business Object Name: CredStoreAdminOperation
Business Object Component Name: Package name: com.mslv.oms.metadatahandler.operation
Description: This new class is used to configure the JPS policy for your custom credential store map and to manage credentials in the credential store.

Attributes
- **credStoreName**
  Type: ObjectName
  Description: Mbean object for credential store: JpsJmxConstants.MBEAN_JPS_CREDENTIAL_STORE
- **globalPolicyName**
  Type: ObjectName
  Description: Mbean object for global policy: JpsJmxConstants.MBEAN_JPS_ADMIN_POLICY_STORE

Business Object Operations

**configJPSPolicy**
This method is used to:
- Update the JPS policy to use the default credential store map (the default map is not configured during OSM installation).
- Configure the JPS policy with an entry for your custom credential store map. The supported operations are add and remove.

Example credential data in XML file with JPS policy information:

```
......
    <jpspolicy operation="add">
      <mapname>osm_systemAmap</mapname>
    </jpspolicy>
```

**configCredentialStore**

This method is used to manage credentials in the WebLogic Server credential store. Use this command to manage credentials of external systems. Use the "userAdmin Command" to configure the OSM user in the credential store.

Example credential data in the XML file with credential information:

```xml
<credential operation="create">
  <mapname>osm_systemAmap</mapname>
  <keyname>user1</keyname>
  <user>mobileUser1</user>
  <password>user1pwd</password>
</credential>
```

Supported operations are **create**, **update**, and **delete**.

**Note:** If **create** fails when specified map/key values already exist in the credential store, set attribute "overwrite" to "false".

**Note:** Password value can be provided through console input.
CredStore

Credential store object.

The credential store object is the domain credential store class which contains a single instance of the CredentialStore object. The JpsServiceLocator APIs in the Credential Store Framework look up the single instance of the CredentialStore object.

Package name: com.mslv.oms.security.credstore

Attributes

Name: store
Type: Oracle.security.jps.service.credstore.CredentialStore
Description: A reference object to the JPS credential store object.

Business Object Operations

getInstance
Description: Return an instance of the object. Only a single instance of the class is ever created. If "store" is not initiated, look up the credential store from class "oracle.security.jps.service.credstore.CredentialStore".

Operation Outputs: Output Name: store; Type: CredStore; Description: An instance of the CredentialStore object.

getJPSCredentialStore
Description: Retrieving attribute "store".

Operation Outputs: Output Name: store; Type: oracle.security.jps.service.CredentialStore.

Output of new methods

An instance of the object is returned by getInstance(). At the first time invocation, object will be initiated, and a credential store of class Oracle.security.jps.service.credstore.CredentialStore is resolved through the CSF lookup API.

Error Conditions

Improper JPS configuration can cause credential store lookup to fail.

Usage Notes

This API can be used directly if you have your own implementation JAVA class of "ViewRuleContext" and "AutomationContext".
PasswordCredStore

Password credential store object.

Use com.mslv.oms.security.credstore.PasswordCredStore APIs in your JAVA classes to retrieve user name and password from the credential store.

Package Name

com.mslv.oms.security.credstore

Attributes

- **credstore**
  Type: CredStore
  Description: A reference object to OSM credential store object.

- **OSM_CREDENTIAL_MAPNAME**
  Type: String (static final)
  Sensitive: Value is "osm"
  Description: Pre-defined map name for OSM application in credential store.

- **OSM_CREDENTIAL_KEYNAME_PREFIX**
  Type: String (static final)
  Sensitive: Value is "osmUser_"
  Description: Prefix of key names used for OSM users in credential store.

Business Object Operations

**Operation Name: getPasswordCredential**

Description

Return a PasswordCredential object stored with specified map and key names.

**Input Parameters**

- **mapName**
  Type: String
  Description: Map name of the stored password credential object

- **keyName**
  Type: String
  Description: Key name of the stored password credential object

**Operation Outputs**

- **passwordCredential**
  Type: PasswordCredential
  Description: An object of Oracle.security.jps.service.credstore.PasswordCredential, which contains credential information stored under map and key name pair.

**Operation Name: getCredential**

Description

Return a string of user name and password for specified map and key names.
Input Parameters

mapName
Type: String
Description: Map name of the stored password credential object

keyName
Type: String
Description: Key name of the stored password credential object

Operation Outputs

Type: String
Description: A string contains user name and password information stored under map and key name pair. Format is "username/password".

Operation Name: getOsmCredentialPassword
Description
Return password value for specified OSM user. This API is used to access credentials stored in the credential store using the default map and key names that follow OSM naming convention:

- Map name is osm
- Key name is osmUser_+username

Input Parameters

username
Type: String
Description: OSM user name.

Operation Outputs

Type: String
Description: A string contains password value for specified OSM user. OSM user name and password values are stored under credential store with map value OSM_CREDENTIAL_MAPNAME, and key value starts with OSM_CREDENTIAL_KEYNAME_PREFIX, following with user name.

Operation Name: getCredentialAsXML
Description
Return user name and password in XML format for specified map and key names.

Input Parameters

mapName
Type: String
Description: Map name of the stored password credential object

keyName
Type: String
Description: Key name of the stored password credential object

Operation Outputs

Type: org.w3c.dom.Element
Description: An element that contains user name and password information stored under map and key name pair.
Output of Methods
These methods will return a PasswordCredential/String/Element object if the credential store contains a credential with specified map name and key name. If a match is not found, null value will be returned.

Error Conditions
Improper JPS configuration can cause “read” operation on the credential store to fail due to “no permission” error. Incorrect map and key names can cause “no credential found” problem.

Usage Notes
This API can be used directly if you have your own implementation JAVA class of "ViewRuleContext" and "AutomationContext".

Example: Retrieve Password from OSM Default Map Given User Name
```java
PasswordCredStore pwdCredStore;
try {
    pwdCredStore = new PasswordCredStore();
    return pwdCredStore.getOsmCredentialPassword(username);
} catch (final Exception e) {
    throw new AutomationException("Fail to find password credential with specified map and key name.", e);
}
```

Example: Retrieve Password from Custom Map Given Map and Key Names Used to Store the Credentials
```java
PasswordCredStore pwdCredStore;
try {
    pwdCredStore = new PasswordCredStore();
    return pwdCredStore.getCredentialAsXML(map, key);
} catch (final Exception e) {
    throw new AutomationException("Fail to find password credential with specified map and key name.", e);
}
```
CredStoreException

Credential store exception object.

Package Name

com.mslv.oms.security.credstore

Attributes

Name: target
Type: Exception
Description: Target exception is the original exception caught in the three OSM credential store classes: CredStore, PasswordCredStore, JPSPasswordCredential.

Business Object Operations

Operation Name: getTargetException
Description
Get attribute "target".

Operation Outputs
exception
Type: Exception

Usage Notes

This API can be used directly if you have your own implementation JAVA class of "ViewRuleContext" and "AutomationContext".
SoapAdapter

Use the attributes for the credential store when you define data provider instances in your cartridges.

For detailed information on data provider adapters, see the discussion on behaviors in OSM Developer’s Guide.

Description

Built-in adapter.

Attributes

- CREDENTIAL_MAPNAME_PARAM
  Type: String
  Description: Defines the parameter name to be specified in data provider for SOAP. A constant with value "oms:credentials.mapname".

- CREDENTIAL_KEYNAME_PARAM
  Type: String
  Description: Defines the parameter name to be specified in data provider for SOAP. A constant with value "oms:credentials.keyname".

Business Object Operations

Operation Name: retrieveInstance

Description
This method includes support to retrieve credential information from the credential store, from map and key name parameters if provided.

Business Logic
The business logic for retrieveInstance is as follows:

- If "oms:credentials.username" is provided in parameters:
  - If "oms:credentials.password" is also provided in parameter, then input values are used directly.
  - If "oms:credentials.password" is not provided in the parameter, call context API "getOsmCredentialPassword(username)" to retrieve the password value from the credential store and use it in the SOAP request.

- Otherwise, if "oms:credentials.mapname" and "oms:credentials.keyname" are provided in the parameters, call context API "getCredential(mapname, keyname)" to retrieve user name and password, and use them in the SOAP request.

Error Conditions

Invalid map and key names can cause credential lookup to return a "null" object.
Message text is "Password credential with map name %s and key name %s does not exist in the credential store."

Usage Notes

Do not use operation APIs directly in this object.
ObjectelHPPTAdapter

Use the attributes for the credential store when you define data provider instances in your cartridges.

For detailed information on data provider adapters, see the discussion on behaviors in OSM Developer’s Guide.

Description

Built-in adapter. Objectel HTTP adapter.

Attributes

- CREDENTIAL_MAPNAME_PARAM
  Type: String
  Description: Defines the parameter name to be specified in data provider for Objectel HTTP type. A constant with value "obj:mapname".

- CREDENTIAL_KEYNAME_PARAM
  Type: String
  Description: Defines the parameter name to be specified in data provider for Objectel HTTP type. A constant with value "obj:keyname".

- mapname
  Type: String
  Description: Value specified for map name parameter.

- keyname
  Type: String
  Description: Value specified for key name parameter.

Business Object Operations

Operation Name: parseParameters
Description
This method includes support to parse parameters for credential store map and key names. Add context to input parameter. Same method in the super class will be changed as well.

Input Parameters
Context
Type: ViewRuleContext

Operation Name: sendCommand
Description
This method includes support to retrieve credential information from the credential store, from map and key name parameters if provided.

Business Logic
The business logic for sendCommand is as follows:

- If "obj.user_name" is provided in the parameters:
If "obj:password" is also provided in the parameter, then input values are used directly.

If "obj:password" is not provided in the parameter, call context API "getOsmCredentialPassword(username)" to retrieve password value from the credential store and use it in the SOAP request.

- Otherwise, if "obj:mapname" and "obj:filename:" are provided in parameters, call context API "getCredential(mapname, keyname)" to retrieve user name and password and use them in the SOAP request (after the command, the code will send a SOAP message via HTTP to the specified URL).

Usage Notes
Do not use operation APIs directly in this object.

Error Conditions
Invalid map and key names can cause credential lookup to return a "null" object.
Message name: ViewRuleFailedException
Message text: "Password credential with map name %s and key name %s does not exist in the credential store."
**ViewRuleContext**

Use operation APIs defined in this interface object for the credential store.

**Description**

Interface object.

**Business Object Operations**

**Operation Name: getCredential**

**Description**

Return a string of user name and password for specified map and key names.

**Input Parameters**

- **map**
  - Type: String
  - Description: Map name

- **key**
  - Type: String
  - Description: Key name

**Operation Outputs**

- **Type**: String
  - Description: A string contains user name and password information stored under map and key name pair. Format is "username/password".

Details on operation getCredential():

```java
/** *
 * Get user name and password values in string format from credential store, given map and key values.
 * @param map Map name of the credential stored in domain credential store.
 * @param key Key name of the credential stored in domain credential store.
 * @return A String that contains user name and password values, separated by "/".
 * @throws CredStoreException If the application cannot access credential store, or if there is no permission to read the credential store with given map and key values, or if the credential is expired.
 */
String getCredential(final String map, final String key) throws TransformerException;
```

**Operation Name: getOsmCredentialPassword**

**Description**

Return password value for specified OSM user. This API is used to access credentials stored in the credential store using the default map and key names that follow OSM naming convention:

- Map name is `osm`
Key name is `osmUser_+username`

**Input Parameters**

`username`
Type: String
Description: OSM user name.

**Operation Outputs**
Type: String
Description: Return password value for specified OSM user. OSM user name and password values are stored under credential store with map value `OSM_CREDENTIAL_MAPNAME`, and key value starts with `OSM_CREDENTIAL_KEYNAME_PREFIX`, following with user name.

**Error Conditions**

Improper JPS configuration can cause creation of PasswordCredStore to fail.
Message Name: ViewRuleFailedException
Message Text: "Fail to create PasswordCredStore."

**Usage Notes**

This API is often used in XQuery scripts.
AutomationContext

Use operation APIs from AutomationContext interface to retrieve credentials in XQuery code for automation tasks.

See "Example: Retrieve Password from OSM Default Map Given User Name".

See "Example: Retrieve Password from Custom Map Given Map and Key Names Used to Store the Credential".

Description

Interface object.

Business Object Operations

Operation Name: getCredentialAsXML

Description

Get user name and password values in XML format given map and key values of the credential.

Input Parameters

map
Type: String
Description: Map name

key
Type: String
Description: Key name

Operation Outputs

Type: org.w3c.dom.Element
Description: An element that contains user name and password information stored under map and key name pair.

Details on operation getCredentialAsXML():

/**
 * Get user name and password values in XML format given map and key values of the credential.
 * @param map
 * Map name of the credential stored in domain credential store.
 * @param key
 * Key name of the credential stored in domain credential store.
 * @return User name and password for the user in this XML format:
 *     <Credential xmlns="urn:com:metasolv:oms:xmlapi:1">
 *         <Username>NAME</Username>
 *         <Password>PASSWORD</Password>
 *     </Credential>
 * @throws CredStoreException
 * If the application cannot access credential store, or if there is no permission to read the credential store with given map and key values, or if the credential is expired.
 */
Document getCredentialAsXML(final String map, final String key) throws CredStoreException
Operation Name: getOsmCredentialPassword
Description
Return password value for specified OSM user. This API is used to access credentials stored in the credential store using the default map and key names that follow OSM naming convention:

- Map name is osm
- Key name is osmUser_+username

Input Parameters
username
Type: String
Description: OSM user name.

Operation Outputs
Type: String
Description: Password value for specified OSM user. OSM user name and password values are stored under credential store with map value OSM_CREDENTIAL_MAPNAME, and key value starts with OSM_CREDENTIAL_KEYNAME_PREFIX, following with user name.

Error Conditions
Fail to read credential store due to improper JPS configuration or invalid map and key names.

Message Name: AutomationException
Message Text: "Fail to create PasswordCredStore. Password credential with map name %s and key name %s does not exist in the credential store."

Example: Retrieve Password from OSM Default Map Given User Name
```
declare variable $context external;
let $osmPwd := context:getOsmCredentialPassword($context, $username)
```

Example: Retrieve Password from Custom Map Given Map and Key Names Used to Store the Credential
```
declare namespace oms="urn:com:metasolv:oms:xmlapi:1";
declare variable $context external;
let $customCred := context:getCredentialAsXML($context, "osmTest", $username)/oms:Credential
let $customerName := $customCred/oms:Username/text()
let $customPwd := $customCred/oms:Password/text()
```

Note: This example assumes your map name is (osmTest).
This chapter describes how to change the appearance and functionality of Oracle Communications Order and Service Management (OSM) GUI applications.

About Configuring the User Experience

OSM supports the customization of the information presented to and collected from Web Client users to match a wide variety of user roles and tasks. This makes it easy to customize the user experience for edits and data validation without having to code Java Server Pages (JSPs).

The Web Client user experience can be customized in various ways:

- Adding default values and calculating values from different fields on the order
- Viewing and selecting data dynamically from a source outside of OSM
- Adding fonts, colors, and other formatting; adding conditional formatting
- Organizing order data in tables and tabs
- Adding your own custom HTML-based online help (with hyperlinks) and tool tips for your users
- Presenting data in one or more presentation languages
- Enforcing formats of input data, required fields and so on
- Hiding and showing fields relative to other fields
- Adding check boxes and radio buttons
- Conditionally making fields read-only or read-write
- Provide custom information, warning, and error messages to the user to help them

About Behaviors

Behaviors provide the mechanism to exercise greater control over validation and presentation of order data to Task, and Order Management Web Client users. (In earlier versions of OSM, this capability was called the View Framework.) Behaviors are used mainly in the context of manual tasks that you manage with the Task Web Client.

There are nine behavior types that enable you to dynamically control a specific aspect of the order data model. (You can also add new behavior types using Oracle Communications Design Studio). The included behavior types are:
- **Calculation**: Computes the value of a field value based on a formula that references order data.
- **Constraint**: Specifies a condition that must be met for the data to be considered valid.
- **Data Instance**: Declares an instance that can be used by other behaviors.
- **Event**: Specifies an action that is performed when data is modified.
- **Information**: Specifies the label, hint, and help information for the element instance.
- **Lookup**: Specifies a set of dynamically generated choices from which you can select.
- **Read Only**: Dynamically determines whether a value can be modified or not.
- **Relevant**: Dynamically determines whether data is visible or hidden.
- **Style**: Specifies the visual appearance of fields.

Behaviors can be created only for manual tasks. They can be created at the data element level (most general), order level (more specific), or task level (most specific). After the behavior is created in Design Studio, you can model the actions you want it to perform through its properties settings.

See *OSM Developer’s Guide* and Design Studio Modeling OSM Processes Help for more information about behaviors.

You can use Design Studio to define additional behaviors.
This chapter explains how to configure Oracle Communications Order and Service Management (OSM) using `oms-config.xml` and provides a detailed reference of available parameters.

**Working with oms-config.xml**

Various OSM parameters can be configured in `oms-config.xml`. The file can be stored in `oms.ear`, or it can be located externally.

By default, OSM uses `oms-config.xml` packed in `oms.ear` for configuration parameters. To alter the stored `oms-config.xml` file, unpack `oms.ear`, edit `oms-config.xml`, save your changes, and repack `oms.ear`. Redeploy the updated `oms.ear` for changes to take effect.

Oracle recommends the using an external `oms-config.xml` file:

- It is quicker to make changes to an external `oms-config.xml` file because you do not have to unpack and undeploy the old `oms.ear` file and repack and redeploy the updated `oms.ear` file.
- OSM monitors the external `oms-config.xml` file. If any changes are made to it when OSM is running, it automatically reloads `oms-config.xml` and refreshes metadata.

**Using an External oms-config.xml File**

To use an external `oms-config.xml` file:

1. If you have not already done so, unpack `oms.ear` as described in "Using an oms-config.xml File Stored in oms.ear".
2. Make a copy of the `oms-config.xml` file and store it in a location of your choosing.
3. Edit `domain_home/bin/startWebLogic.sh` (`startWebLogic.bat` on Windows) and locate the line that starts WebLogic. It will appear similar to the following:
   ```
   ${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME} -Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy ${JAVA_OPTIONS} ${PROXY_SETTINGS} ${SERVER_CLASS}
   ```
4. Add the following parameter:
   ```
   -Dmslv.oms.config=omsconfigxml_path/oms-config.xml
   ```
   where `omsconfigxml_path` is the path to your customized `oms-config.xml` file.
   The updated line will appear similar to the following:
Working with oms-config.xml

```
${JAVA_HOME}/bin/java ${JAVA_VM} ${MEM_ARGS} -Dweblogic.Name=${SERVER_NAME} -Djava.security.policy=${WL_HOME}/server/lib/weblogic.policy -Dmslv.oms.config=/Users/wls/oms-config.xml ${JAVA_OPTIONS} ${PROXY_SETTINGS} ${SERVER_CLASS}
```

5. Save and close the file.

Using an oms-config.xml File Stored in oms.ear

OSM run-time parameters are configured in oms-config.xml.

To alter oms-config.xml, unpack oms.ear, edit oms-config.xml, and repack oms.ear.

To unpack and repack oms.ear, use the following batch files and scripts:

- **UNIX:**
  unpackOSM.sh and packOSM.sh. The files are in OSM_Home/SDK/Customization

- **Windows:**
  unpackOMS.bat and packOMS.bat. The files are in OSM_Home\SDK\Customization

The SDK directory exists only if you installed the SDK during a **Custom** installation.

---

**Important:** You can install only the SDK on a Windows or UNIX machine by running the OSM installer again, and selecting **Custom** installation.

---

To configure OSM by editing the oms-config.xml file:

1. Copy the oms.ear file from the production system to the system containing the pack and unpack scripts or batch files.

---

**Note:** Oracle recommends that you make a backup copy of your original oms.ear file, in case you need to restore the default settings.

---

2. Go to OSM_Home/SDK/Customization.

3. Do one of the following:

- **UNIX:**
  Edit unpackOMS.sh and packOMS.sh and set JAVA_HOME variable to the path to the JDK on your UNIX machine.

- **Windows:**
  Edit unpackOMS.bat and packOMS.bat and set JAVA_HOME variable to the path to the JDK on your Windows machine.

---

**Note:** If the path in JAVA_HOME contains a space, enclose the path in quotation marks. For example:

```
set JAVA_HOME="C:\oracle\middleware test\jdk160_11"
```

---

4. Do one of the following:
Configuring OSM with oms-config.xml

5. Edit `osm-ejb\security\META-INF\oms-config.xml` to customize the parameters you want to modify. See Table 6–1 for a list of parameters in `oms-config.xml`.

6. Save your changes.

7. Do one of the following:
   - UNIX: Run `packOMS.sh`, which repacks `oms.ear`.
   - Windows: Run `packOMS.bat`, which repacks `oms.ear`.
     This creates a new `oms.ear` file.

8. Copy the updated `oms.ear` to a directory on the host running the OSM server, for example, `OSM_home/bin`.

9. Log in to the WebLogic Administration Console.

10. In the Domain Structure pane, click Deployments.

11. In the Deployments window, select the box next to `oms.ear`, click Stop, and select the option appropriate for your situation.

12. With the box next to `oms.ear` selected, click Delete.

13. Click Activate Changes.

14. Click Install.

15. In the Install Application Assistant window, browse to the location of the updated `oms.ear` file and click Finish.

16. Select the box next to `oms.ear` and click Start and Servicing all Requests.

### oms-config.xml Parameters

Table 6–1 describes the parameters that can be configured in the `oms-config.xml` file.

---

**Important:** Oracle recommends that you make a backup copy of `oms-config.xml` before making any changes to it.
### Table 6–1 Description of oms-config.xml Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Min</th>
<th>Max</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>max_worklist_rows</td>
<td>Maximum number of rows shown in the worklist</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Changing this parameter can affect system performance.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oracle recommends that you not change this parameter to a value significantly larger than the default.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_notification_rows</td>
<td>Maximum number notifications returned in one request in the Task Web client</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Changing this parameter can affect system performance.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oracle recommends that you not change this parameter to a value significantly larger than the default.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>max_query_rows</td>
<td>Maximum number of rows returned in the Query Results page in the Task Web client</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Changing this parameter can affect system performance.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Oracle recommends that you not change this parameter to a value significantly larger than the default.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>worklist_page_size</td>
<td>Maximum number of rows shown on each Worklist page in the Task Web client</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>15</td>
</tr>
<tr>
<td>userlist_page_size</td>
<td>Maximum number of rows shown on each User List page in the Task Web client</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>15</td>
</tr>
<tr>
<td>notifications_page_size</td>
<td>Maximum number of lines shown on the Notifications page in the Task Web client</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>15</td>
</tr>
<tr>
<td>query_results_page_size</td>
<td>Maximum number of lines shown on one page in the Query Results page in the Task Web client</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>15</td>
</tr>
<tr>
<td>jdbc_fetch_size</td>
<td>JDBC fetch size</td>
<td>integer</td>
<td>1</td>
<td>64000</td>
<td>200</td>
</tr>
<tr>
<td>url_root</td>
<td>The base URL for the OSM Web applications</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>oms</td>
</tr>
<tr>
<td>email_server</td>
<td>IP address or server name of the email server used to send OSM notifications</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>127.0.0.1</td>
</tr>
<tr>
<td></td>
<td>Oracle recommends that you not change this parameter from the installation setting.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email_server_port</td>
<td>Port number of the email server used to send OSM notifications</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>Oracle recommends that you not change this parameter from the installation setting.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Min</td>
<td>Max</td>
<td>Default</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----</td>
<td>---------</td>
</tr>
<tr>
<td>admin_email_address</td>
<td>Default email address to which to send notifications. The default value set by oms-parameter-default is ignored.</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Oracle recommends that you not change this parameter from the installation setting.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>email_server_ssl</td>
<td>Enables SSL connection to the email server when set to True</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>email_server_authentication</td>
<td>Enables use of SSL authentication when connecting to the email server when set to True</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>remark_change_timeout_hours</td>
<td>Time in hours after the date and time a remark is added to an order that changes are no longer allowed to the remark</td>
<td>integer</td>
<td>1</td>
<td>1000</td>
<td>120</td>
</tr>
<tr>
<td>attachment_file_system_name</td>
<td>Name of the T3 file service configured in WebLogic that manages order attachments for OSM</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>max_attachment_size</td>
<td>Maximum size in MB of documents attached to orders in the Task Web client Changing this parameter impacts OSM WebLogic host machine file system usage.</td>
<td>integer</td>
<td>1</td>
<td>256</td>
<td>10</td>
</tr>
<tr>
<td>database_timezone_offset</td>
<td>Offset in seconds of the database server’s time zone. The database time zone offset is used to calculate the time zone of the database, which may be different from the time zone of the application. The maximum offset is 14 hours. The default value is ignored: this parameter is defaulted to the database server’s time zone offset. It is not recommended to change this parameter from the installation setting.</td>
<td>integer</td>
<td>-50400</td>
<td>50400</td>
<td>N/A</td>
</tr>
<tr>
<td>max_read_only_field_length</td>
<td>Maximum length of a read-only field in the Order Editor</td>
<td>integer</td>
<td>1</td>
<td>512</td>
<td>30</td>
</tr>
<tr>
<td>create_order_show_namespace</td>
<td>If set to False, no list is displayed in the Task Web client to select a namespace when creating an order</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>load_users_from_database</td>
<td>In the XML API, you can call ListUsers. This tells the handler to either load the users from the database or from the J2EE security server (WebLogic).</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>False</td>
</tr>
</tbody>
</table>
Table 6–1  (Cont.) Description of oms-config.xml Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Min</th>
<th>Max</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>workstream_refresh_interval</td>
<td>In the Task Web client, the number of milliseconds to wait before attempting to retrieve the next available task in a workstream. It is not recommended to change this parameter from the installation setting. Changing this parameter can affect system performance.</td>
<td>integer</td>
<td>500</td>
<td>60000</td>
<td>1000</td>
</tr>
<tr>
<td>max_workstream_retry</td>
<td>In the Task Web client, the number of retries to attempt to retrieve the next available task in a workstream before redirecting the user back to the Worklist screen. It is not recommended to change this parameter from the installation setting. Changing this parameter can affect system performance.</td>
<td>integer</td>
<td>1</td>
<td>1000</td>
<td>30</td>
</tr>
<tr>
<td>workstream_predefined_</td>
<td>Controls how task completion status buttons are displayed in the workstream order editor. If set to True, predefined task statuses are displayed on a separate line in their predefined order. User-defined statuses are displayed on the next line. The display order of user-defined statuses is controlled by the model designer. If set to False, all task completion statuses are displayed in a single line. The display order of all task completion statuses is controlled by the model designer.</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>status_display_fixed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hide_dirty_order</td>
<td>Normally, after a task is completed and you return to the worklist, the task is displayed in bold italics to indicate it has been completed. If set to True, the completed task will not be displayed in the worklist.</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>False</td>
</tr>
<tr>
<td>disable_edit_on_server_</td>
<td>If set to True, if a server refresh occurs while an order is being edited, the edit function becomes disabled.</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>refresh</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>login_screen</td>
<td>Initial screen displayed when the Task Web client is started. Valid values include: about, default, home, query, worklist</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>worklist</td>
</tr>
</tbody>
</table>
### Table 6–1 (Cont.) Description of oms-config.xml Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Min</th>
<th>Max</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>auto_logout_warning_offset_minutes</td>
<td>Number of minutes prior to session timeout (auto-logout) to display a warning message to the user. A value of -1 indicates no warning is issued. The maximum value is 1440 minutes (24 hours). It is not recommended to change this parameter from the installation setting.</td>
<td>integer</td>
<td>1</td>
<td>1440</td>
<td>5</td>
</tr>
<tr>
<td>com.mslv.oms.handler.order.cache.OrderCacheManager</td>
<td>Cache manager</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>com.mslv.oms.handler.order.cache.jboss.JBossOrderByCacheManager</td>
</tr>
<tr>
<td>com.mslv.oms.util.xml.XMLHelper.DocumentBuilderFactory</td>
<td>Name of the configuration file containing metadata definitions for a custom menu action</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>custommenuaction_model_location</td>
<td>Name of the configuration file containing metadata definitions for a custom menu action</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>event_poller_interval</td>
<td>Time in seconds OSM waits before polling for new events</td>
<td>integer</td>
<td>100</td>
<td>60000</td>
<td>1000</td>
</tr>
<tr>
<td>event_poller_mutex_timeout</td>
<td></td>
<td>integer</td>
<td>0</td>
<td>60000</td>
<td>10000</td>
</tr>
<tr>
<td>com.mslv.oms.security.FactoryRegistry.HandlerFuncFactory</td>
<td>Name of the process that manages the handling of high activity orders in clustered systems</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>com.mslv.oms.security.FactoryRegistry.HandlerFuncFactory</td>
</tr>
<tr>
<td>com.mslv.oms.handler.cluster.ClusteredHandlerFactory.HighActivityOrder.CollectionCycle.Enabled</td>
<td>Enables high-activity order collection cycles</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>com.mslv.oms.handler.cluster.ClusteredHandlerFactory.HighActivityOrder.CollectionCycle.InitialDelay</td>
<td>The amount of time in milliseconds to wait before the first collection cycle. This wait period allows servers to start and the system to stabilize before statistics used to determine high activity orders are collected.</td>
<td>integer</td>
<td>1000</td>
<td>360000</td>
<td>10000</td>
</tr>
<tr>
<td>com.mslv.oms.handler.cluster.ClusteredHandlerFactory.HighActivityOrder.RequestPerSecondThreshold</td>
<td>The number of requests per second per order that must be processed for an order to be considered a high-activity order and eligible for special load balancing</td>
<td>integer</td>
<td>1</td>
<td>1000</td>
<td>50</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Min</td>
<td>Max</td>
<td>Default</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>com.mslv.oms.security.HandlerCallbackFactoryRegistry.HandlerCallbackFactory</td>
<td></td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>com.mslv.oms.security.HandlerCallbackFactory</td>
</tr>
<tr>
<td>com.mslv.oms.security.OrderViewAccessProvider</td>
<td>Registers a security callback for the OrderViewAccessException exception</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>com.mslv.oms.cartridgemgmt.DeployCartridgeMDB.CartridgeDeploymentTransactionTimeout</td>
<td>Default transaction timeout interval in seconds for Oracle Communications Design Studio cartridge deployment. Design Studio can override the default value through the environment property.</td>
<td>integer</td>
<td>100</td>
<td>3600</td>
<td>600</td>
</tr>
<tr>
<td>com.mslv.oms.cartridgemgmt.cache.DeployCartridgeCache.DeployCartridgeRequestTimeToLiveMinutes</td>
<td>Default eviction timeout interval in minutes for Design Studio cartridge deployment requests to be cleaned up from the cache. Design Studio can override the default value through the environment property.</td>
<td>integer</td>
<td>5</td>
<td>360</td>
<td>30</td>
</tr>
<tr>
<td>com.mslv.oms.model.transform.OrderTransformer.ModelURL</td>
<td></td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.cluster.BusinessRequestBalancer.OrderRelease.Timeout</td>
<td>The time in seconds to wait for a non-exclusive lock to be placed on an order. A non-exclusive lock is required prior to OSM attempting to process an order. Exclusive locks are acquired by OSM when an order is about to be ejected from the order cache or when an order is being transferred from one node in an OSM cluster to a different node. Exclusive locks prevent non-exclusive locks from being acquired.</td>
<td>integer</td>
<td>200</td>
<td>500</td>
<td>200</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.cluster.BusinessRequestBalancer.ServerStateScanning.Interval</td>
<td>Time in milliseconds to wait between scanning server states. This value is used to determine how frequently the WebLogic server is checked to see if it is in a RUNNING state prior to enabling application services such as intracluster communication or the JMS server. These services cannot be safely enabled until after the WebLogic server is in a RUNNING state.</td>
<td>integer</td>
<td>1000</td>
<td>10000</td>
<td>5000</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.orchestration.generation.CreateOrder</td>
<td>Default Orchestration plan XQuery to create an order update. This parameter is not intended to be end-user adjustable. Oracle recommends not changing this parameter.</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>oracle/communications/ordermanagement/orchestration/generation/CreateOrder.xquery</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Min</td>
<td>Max</td>
<td>Default</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>------</td>
<td>------</td>
<td>----------------------</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.orchestration.generation.ControlDataLocation</td>
<td>Specifies the node in an orchestration order’s order template that contains control data and order items. The data at this location is automatically populated by OSM when the orchestration plan is generated.</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>ControlData/OrderItem</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.util.net.CatalogUriResolver.DefaultXmlCatalogsUris</td>
<td>List of URLs specifying the XML Catalogs that are used system-wide. On all OS platforms, entries are separated by a semicolon (;).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.config.ModelResourceClasspath</td>
<td>List of URIs specifying either JAR files or directories containing class files that will be made available to the OSM class loader. On all OS platforms, entries are separated by a semicolon (;).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>com.mslv.oms.automation.AutomationDispatcher.DefaultAutomationPluginDispatchMode</td>
<td>The dispatch mode to use by default for all automation plug-ins. LEGACY: run in external EAR file INTERNAL: run in oms.ear</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>INTERNAL</td>
</tr>
<tr>
<td>is_tablelayout_height_fixed</td>
<td>If set to True, the table height equals the value of height_of_tablelayout. If set to False, the table height adjusts according to the number of rows in the table.</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>height_of_tablelayout</td>
<td>Height in pixels of the table layout</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>400px</td>
</tr>
<tr>
<td>no_of_rows_in_textarea_without_scroll</td>
<td></td>
<td>integer</td>
<td>3</td>
<td>1000</td>
<td>3</td>
</tr>
<tr>
<td>max_no_of_rows_in_textarea_with_scroll</td>
<td></td>
<td>integer</td>
<td>3</td>
<td>1000</td>
<td>3</td>
</tr>
<tr>
<td>com.mslv.oms.handler.completeorder.CompleteOrderHandlerEJB.OrchPlanLock.Timeout</td>
<td>Time in seconds that OSM will wait while trying to acquire an exclusive lock on an orchestration plan. This lock is required only when OSM detects that all order components within the orchestration plan have completed and the order can complete.</td>
<td>integer</td>
<td>1</td>
<td>3600</td>
<td>1</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.orchestration.generation.DumpOrchestrationPlan</td>
<td>If set to True, OSM saves a copy of every generated orchestration plan in XML format to the file orderId_orderType_orchestrationSequence_orchestrationPlanOutput.xml. Note: This parameter should be used only at the request of Oracle Support for diagnostic purposes.</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>False</td>
</tr>
</tbody>
</table>
**normalize_space_for_lookup_rule**

Determines whether spaces are normalized within XML values that are used in the results of a Lookup behavior. If set to **True**, the results are normalized by trimming leading and trailing spaces and replacing repeating spaces with a single space, in accordance with the normalize-space XPath function as defined here: [http://www.w3.org/TR/xpath/#func tion-normalize-space](http://www.w3.org/TR/xpath/#function-normalize-space).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Min</th>
<th>Max</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>webui_order_info_pane_order_item_sort_ascending</code></td>
<td></td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td><code>webui_order_info_pane_order_component_sort_ascending</code></td>
<td></td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td><code>oracle.communications.ordermanagement.cache.UserPreferenceCache</code></td>
<td>Specifies the name of the Coherence cache configuration to use for user preference information. By default, a local cache is used for non-clustered environments. For clustered environments a “near cache” is used to ensure changes to user preference information is automatically synchronized between cluster nodes.</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>near</td>
</tr>
<tr>
<td><code>oracle.communications.ordermanagement.RuleDelayTaskPoller.MaxRuleTaskProcessors</code></td>
<td>Specifies the maximum number of rule task processors</td>
<td>integer</td>
<td>1</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td><code>oracle.communications.ordermanagement.RuleDelayTaskPoller.MaxDelayTaskProcessors</code></td>
<td>Specifies the maximum number of delay task processors</td>
<td>integer</td>
<td>0</td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td><code>oracle.communications.ordermanagement.RuleDelayTaskPoller.Interval</code></td>
<td>Specifies the pause time in milliseconds between consecutive executions of the rule and delay task processors.</td>
<td>integer</td>
<td>1000</td>
<td>60000</td>
<td>5000</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Min</td>
<td>Max</td>
<td>Default</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>---------</td>
</tr>
</tbody>
</table>
| oracle.communications.ordermanagement.security.GlobalQueryRoles    | An order type/source can be queried if the user has one of the following permissions on the order type/source:  
  - Permission to create the order (creation task)  
  - Permission on at least one task  
  - Permission on at least one flex header where the flex header node is in that order’s order template  
  - At least one query view assigned.  
  The workgroups where one of the previous permissions is set must all have the Search View permission assigned. Users can always see default and defaultOrchestration cartridges. For backwards compatibility, this configuration option can be used to force this function to return all order type/source in the system. The workgroup names should be separated by a semicolon, a comma, and a colon (`;:`).  
  This parameter makes the OSM consistent with pre 7.0.3.1 behavior. | string | N/A   | N/A   | N/A     |
| oracle.communications.ordermanagement.order.UseUnionOfFiltersAcrossRoles | Filters are OR’ed across roles and the results are a union of data across all the results so that if a user is a member of a workgroup with privileges on an order within a cartridge and that workgroup has no filters, the user can see all orders in that cartridge. To see all the results AND’ed so that an intersection of the results is obtained, the names of the specific workgroup should be given in the configuration file separated by a semicolon, a comma, and a colon (`;:`).  
  This parameter has been added for backward compatibility with older versions of OSM. | string | N/A   | N/A   | N/A     |
<p>| order_editor_submit_mode_threshold                                   | The response time of the Order Editor page increases with the number of nodes in an order. To avoid slow response times, the order_editor_submit_mode_threshold parameter is configured to a threshold value for node instances saved in the system. If the number of saved instances increases this threshold value, the system automatically switches from AJAX to form-submit mode when edited orders are saved or submitted for processing. | integer| 1     | 9999999| 15      |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Min</th>
<th>Max</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>time_out_override_for_jms_adapter</td>
<td>The amount of time to wait before throwing the JMS response timeout.</td>
<td>integer</td>
<td>15000</td>
<td>300000</td>
<td>15000</td>
</tr>
<tr>
<td>oracle.communications.order_management.amendment.DataEnrichmentAware</td>
<td>Determines whether OSM is aware of changes to order data from Task Web client users assigned to manual tasks or from automation plug-ins. If set to True, OSM compares revision order data to the current order including any changes from manual tasks or automation-plug-ins. If set to False, OSM compares revision orders to the last submitted order data excluding changes from manual tasks or automation plug-ins.</td>
<td>boolean</td>
<td>N/A</td>
<td>N/A</td>
<td>True</td>
</tr>
<tr>
<td>oracle.communications.order_management.table-layout.size</td>
<td>The number of rows displayed in the table in a single view (i.e. without scrolling), can be configured.</td>
<td>integer</td>
<td>0</td>
<td>50</td>
<td>10</td>
</tr>
<tr>
<td>oracle.communications.order_management.table-layout.fetch-size</td>
<td>The number of rows fetched at a time from the server is configurable; which means that not all the rows that are available for the component on the server are fetched and displayed on the client. The number of rows that are displayed on the client are just enough to fill the viewport. More rows are fetched as the user scrolls the component vertically.</td>
<td>integer</td>
<td>5</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>oracle.communications.order_management.table-layout.threshold</td>
<td>This feature introduced a threshold system parameter above which multi-instance group nodes will automatically be displayed as a table. The system parameter will be read from the existing oms-config.xml configuration file at application startup and metadata refresh, as with current system parameters. A non-integer value or a negative value disables this feature.</td>
<td>integer</td>
<td>N/A</td>
<td>N/A</td>
<td>50</td>
</tr>
<tr>
<td>oracle.communications.order_management.security.access.summary</td>
<td>Grants access to the Summary tab in the Order Management Web client, by workgroup names. Only users of specific workgroups can view the Summary tab. Separate values with commas, semicolons, or colons. Values can contain the asterisk wildcard character, where the asterisk can match a string of characters (for example, a value of user* matches workgroup names user1, user2, user3, and so on).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>* (meaning all workgroups)</td>
</tr>
<tr>
<td>Name</td>
<td>Description</td>
<td>Type</td>
<td>Min</td>
<td>Max</td>
<td>Default</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>oracle.communications.order management.security.access.data</td>
<td>Grants access to the Data tab in the Order Management Web client, by workgroup names. Only users of specific workgroups can view the Data tab. Separate values with commas, semicolons, or colons. Values can contain the asterisk wildcard character, where the asterisk can match a string of characters (for example, a value of user* matches workgroup names user1, user2, user3, and so on).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>* (meaning all workgroups)</td>
</tr>
<tr>
<td>oracle.communications.order management.security.access.amendments</td>
<td>Grants access to the Amendments tab in the Order Management Web client, by workgroup names. Only users of specific workgroups can view the Amendments tab. Separate values with commas, semicolons, or colons. Values can contain the asterisk wildcard character, where the asterisk can match a string of characters (for example, a value of user* matches workgroup names user1, user2, user3, and so on).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>* (meaning all workgroups)</td>
</tr>
<tr>
<td>oracle.communications.order management.security.access.orchestration-plan</td>
<td>Grants access to the Orchestration Plan tab in the Order Management Web client, by workgroup names. Only users of specific workgroups can view the Orchestration Plan tab. Separate values with commas, semicolons, or colons. Values can contain the asterisk wildcard character, where the asterisk can match a string of characters (for example, a value of user* matches workgroup names user1, user2, user3, and so on).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>* (meaning all workgroups)</td>
</tr>
</tbody>
</table>
Table 6–1 (Cont.) Description of oms-config.xml Parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Min</th>
<th>Max</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>oracle.communications.ordermanagement.security.access.dependencies</td>
<td>Grants access to the Dependencies tab in the Order Management Web client, by workgroup names. Only users of specific workgroups can view the Dependencies tab. Separate values with commas, semicolons, or colons. Values can contain the asterisk wildcard character, where the asterisk can match a string of characters (for example, a value of user* matches workgroup names user1, user2, user3, and so on).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>* (meaning all workgroups)</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.security.access.activity</td>
<td>Grants access to the Activity tab in the Order Management Web client, by workgroup names. Only users of specific workgroups can view the Activity tab. Separate values with commas, semicolons, or colons. Values can contain the asterisk wildcard character, where the asterisk can match a string of characters (for example, a value of user* matches workgroup names user1, user2, user3, and so on).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>* (meaning all workgroups)</td>
</tr>
<tr>
<td>oracle.communications.ordermanagement.security.access.data-tree</td>
<td>Controls how users see the Order Info region in the Order Management Web client, by workgroup names. The Order Info region can be seen by all users, but only users of specific workgroups can expand the order item child items. Separate values with commas, semicolons, or colons. Values can contain the asterisk wildcard character, where the asterisk can match a string of characters (for example, a value of user* matches workgroup names user1, user2, user3, and so on).</td>
<td>string</td>
<td>N/A</td>
<td>N/A</td>
<td>* (meaning all workgroups)</td>
</tr>
</tbody>
</table>
Configuring the Rule Engine

This chapter describes how to configure the Rule Engine to improve performance and handle rule-processing errors for Oracle Communications Service and Order Management (OSM).

In most cases, you can use the default configuration for the rule engine.

About Configuring the Rule Engine

The rule engine evaluates rules, event delays, and timer delays to determine when to transition to the next task in a process. The engine is implemented as one or more Task Processor threads running on WebLogic servers. There are two types of task processors: the rule task processor evaluates orders at rules and the delay task processor evaluates orders at delays. You can configure additional task processors to improve performance.

Configuring Rule-Engine Error Handling

You can configure how the OSM rule engine handles errors during processing.

To configure error handling, you use SQL to set a value in the OSM database. Table 7–1 shows the configuration parameters.

The first two parameters enable you to change the behavior of the rule engine when an Oracle internal error (ORA-00600) occurs. When that error occurs during the rule execution, the rule engine immediately tries to re-execute the current rule. If the rule executes, the rule engine continues processing. If the rule does not execute, the rule engine tries to re-execute the current rule (the number of retries is dictated by the value you specify in the rule_retries parameter). If it fails, the rule engine generates an error message. You can clear the error message using the System Events tab in the OSM Administrator application.

When an error occurs during the rule or delay processing, a message is recorded as a system event and the rule or delay is marked as invalid by default. The client has an option not to invalidate the task immediately when an error occurs during processing, but after a specific number of retries. Number of retries is defined by the order_rule_threshold parameter.

Table 7–1  Rule Engine Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>rule_delay</td>
<td>Specifies the delay between retries when an Oracle internal error (ORA-00600) occurs.</td>
<td>N/A</td>
</tr>
</tbody>
</table>
To change value of a parameter

You change these parameters by using SQL to edit the OM_PARAMETER table in the OSM schema:

1. Connect to OSM schema using SQL*PLUS.
2. Execute one of the following:
   - To specify the delay between retries when an Oracle internal error occurs:
     ```sql
     update om_parameter
     set value = 'value'
     where mnemonic = 'rule_delay';
     commit;
     ```
   - To specify the number of retries when an Oracle internal error occurs:
     ```sql
     update om_parameter
     set value = 'value'
     where mnemonic = 'rule_retries';
     commit;
     ```
   - To specify how to handle processing errors:
     ```sql
     update om_parameter
     set value = 'value'
     where mnemonic = 'order_rule_threshold';
     commit;
     ```

Configuring the Rule Engine for Performance

The number and type of task processors used in a given environment is determined at installation. You can also reconfigure them at a later time using the oms-config.xml file. After reconfiguration, restart the WebLogic server instance to apply the changes. The configuration parameters apply only to the WebLogic instance that uses the modified oms-config.xml file (your environment may have multiple WebLogic instances with multiple oms-config.xml files).

There are two types of task processors:

- Rule Task Processor, which evaluates only rules.
- Delay Task Processor, which evaluates only delays.
You may want to reconfigure the number or type of task processors as the amount of data you handle grows.

For rule task processors, the following parameters apply:

- `oracle.communications.ordermanagement.RuleDelayTaskPoller.MaxRuleTaskProcessors`. Specifies the maximum number of rule task processors; minimum value 1; maximum 50; default value is 1.
- `oracle.communications.ordermanagement.RuleDelayTaskPoller.Interval`. Specifies the pause time in milliseconds between two invocations of the rule and delay task processors; minimum value 1000; maximum value 60000; default value is 5000.

For delay task processors, the following parameters apply:

- `oracle.communications.ordermanagement.RuleDelayTaskPoller.MaxDelayTaskProcessors`. Specifies the maximum number of rule task processors; minimum value 0; maximum 50; default value is 1.
- `oracle.communications.ordermanagement.RuleDelayTaskPoller.Interval`. Specifies the pause time in milliseconds between two invocations of the rule and delay task processors; minimum value 1000; maximum value 60000; default value is 5000.

There must be at least one rule task processor running. The number of delay task processors can be 0 or above.

If there is a backlog of rule or delay tasks, you can increase the number of rule or delay task processors.

OSM will adjust the number of rule and delay task processors to use no more than 10% of the connection pool size that is configured for the WebLogic instance. The adjusted numbers are written in the managed server's log file. If the adjusted number of rule and delay task processors does not meet your performance requirement, increase the connection pool size or decrease the parameter `oracle.communications.ordermanagement.RuleDelayTaskPoller.Interval`. 
This chapter describes how to manage partitions in your Oracle Communications Order and Service Management (OSM) database schema.

**About Partitioning Your OSM Database Schema**

Database partitioning increases performance for systems with high order volumes, and in OSM installations which use Oracle Real Application Clusters (RAC). Database partitioning allows tables and indexes to be subdivided into smaller pieces which can be individually managed. This makes it easier to manage large volumes of data. For example, you can purge obsolete data in a partition without affecting data residing in other partitions.

Database partitioning, including the addition of new partitions, can be managed automatically by OSM. However, Oracle recommends that you add partitions manually in production environments, especially if you use Oracle RAC.

OSM tables which accumulate order-related information are range-partitioned by ORDER_SEQ_ID. Range partitions are often sub-partitioned using hash partitioning to reduce I/O contention. Tables are partitioned using equi-partitioning. For more information about the different types of partitioning, see the Oracle Database documentation.

The OSM installer allows you to choose whether or not to enable partitioning. The following values, set on the Database Schema Partition Information installer screen, affect partitions created during installation, as well as partitions created automatically by OSM after installation:

- **Orders per Partition**: specifies the number of orders that will be allowed in a partition
- **Number of Sub-partitions**: specifies the number of hash sub-partitions that will be allowed in a partition

OSM creates the first range partition of each partitioned table during the installation (or upgrade) process. New partitions can be created in different tablespaces. OSM can create partitions automatically, as described in "Creating Partitions Automatically" (recommended only for development and testing environments). You can also add partitions manually, as described in "Adding Partitions with Oracle Scripter".

During installation, you choose tablespaces for Order Data and Order Index. If you choose the same tablespace for both of these, the OSM installer creates local index partitions in the default tablespace. New index partitions will be created in the same tablespace as new table partitions. If you specify a different tablespace for index partitions, the OSM installer creates all index partitions in the index tablespace regardless of where table partitions are created.
About OSM Database Partition Configuration

Optimal settings for the Orders per Partition and Number of Sub-partitions fields are dependent on factors such as data volume, throughput requirements, and maintenance policies.

Typical values for Orders per Partition range between 100000 (100 thousand) and 1000000 (1 million).

Hash sub-partitioning reduces I/O contention and improves performance for some queries through partition pruning. The default Number of Sub-partitions value is 64. A smaller number of sub-partitions is not effective for reducing I/O contention in high volume environments. A very large number of sub-partitions can have an adverse impact on queries that cannot take advantage of partition pruning, and can result in slightly increased CPU usage.

Oracle recommends that you set the Oracle Database initialization parameter `deferred_segment_creation` to true (the default). This setting saves disk space and minimizes the time it takes for OSM to automatically create new partitions. This can help avoid long delays and timeouts in database operations.

Starting with Oracle Database 11.2.0.2, the initial extent size for partitioned tables is 8 MB. If you have many hash sub-partitions, partitioned tables can consume a lot of space even if they contain very little data. For example, even with deferred segment allocation, a table with 64 sub-partitions can quickly expand to 512 MB. Although this is not an issue for production environments, it could be an issue in development or testing environments with limited storage capacity. In those environments, you can use a small number of sub-partitions (for example, 4), or a tablespace with uniform extent allocation and a small extent size (for example, 64 KB).

Oracle recommends that you set the Number of Sub-partitions value to a power of 2. Using other values can result in data skew; data unevenly distributed across partitions.

Oracle Database limits the number of partitions to 1024K - 1. For example, if you enter 64 in the Number of Sub-partitions field, range partitions can be of up to 16 KB in size. With a setting of 100000 in the Order per Partition field, you can keep 1.6 billion orders in the database before you must drop old partitions.

About Range Partitioning Configuration

Range partitioning helps you manage large volumes of data. You can drop old partitions if all orders in a partition are closed. You can drop a partition with no pending orders even if older partitions have pending orders. However, if you have long-lived orders (for example, data service orders that take weeks or months to complete), a large partition size can increase the length of time you must keep partitions before you are able to drop them.

Range partitioning improves performance for some queries through partition pruning, a database technique that allows a query access to a subset of partitions. However, a very large partition size makes partition pruning less effective. A very small partition size result in a large number of partitions. This can have a small negative impact on queries that cannot take advantage of partition pruning and can result in increased CPU usage. For more information about the different types of partitioning, see the Oracle Database documentation.

You can change the initial selections used by the OSM installer by updating the `range_partition_size` and `subpartitions_number` OSM database parameters (see "Partition Configuration Parameters"). Updates to these parameters do not affect existing partitions.
If you use an Oracle RAC database, range partitioning is critical for optimal performance. Each Oracle RAC instance processes orders in a different partition. This approach minimizes cluster waits.

**Database Partitioning and Order IDs**

Each database instance can allocate unique IDs to new orders. OSM allocates a block of unique order IDs for each database instance (for both single instance and Oracle RAC databases). The block size equals the partition size. When a block of order IDs is exhausted, the next block is allocated. If the corresponding partition does not exist, OSM will attempt to create it automatically. Range partitioning works best if the partition size is larger than 50,000 orders.

If you use an Oracle RAC database, OSM order IDs are not assigned sequentially. For example, if the partition size is 1 million orders, the group of WebLogic servers interacting with one Oracle RAC instance could generate order IDs 1000001 to 2000000 while another group of WebLogic servers interacting with a second Oracle RAC instance could generate order IDs 2000001 to 3000000. Use a smaller partition size to avoid large differences in the order ID ranges allocated by different instances.

**Managing Partitions**

The OSM installer automatically creates the first partition for each range-partitioned table. Adding new partitions and dropping old ones are the most common partition maintenance operations.

**Partition Configuration Parameters**

The following configuration parameters affect partition maintenance operations. Updates to these parameters do not affect existing partitions:

- **range_partition_size**: This OSM parameter specifies the number of orders per partition. You can change it with the following SQL command, where \( N \) is the value in quotes (for example ‘100000’).

  ```sql
  update om_parameter set value = \( N \) where mnemonic = 'range_partition_size';
  commit;
  ```

**Note:** OSM uses the **range_partition_size** value to create the partition name. The syntax is `P_` followed by a string of zeros up to the **range_partition_size** value. The string is always 18 character long.

For example, if the range_partition_size were 100000, then the partition name is `P_0000000000000000000000000000100000`.

For every new partition, the name of the next partition is the limit of the previous partition plus the value of the range_partition_size parameter.

For example, if the previous partition were `P_0000000000000000000000000000000000000100000` then the next partition would be `P_00000000000000000000000000000000000000000200000`.

- **subpartitions_number**: This OSM parameter specifies the number of hash sub-partitions. You can change it with the following SQL command, where \( N \) is the new value in quotes (for example ‘32’).

  ```sql
  update om_parameter set value = \( N \) where mnemonic = 'subpartitions_number';
  ```
commit;

- **deferred_segment_creation**: This Oracle database initialization parameter, if set to true (the default), forces the database to wait until the first row is inserted into a table before creating segments for tables and their dependent objects. This parameter saves disk space and minimizes the time it takes to create a partition.

### Adding Partitions with Oracle Scripter

To add range partitions, on the Windows machine where you installed the Database Utilities OSM Component, open the *OSM_Home*\*Database*\*osm\*maintenance\dropPartitions\new_part.sql* script using Oracle Scripter and update the values of the parameters in Table 8–1. The limit of the next partition is the limit of the latest partition plus the value of the **range_partition_size** parameter.

---

**Note**: To specify the tablespace for the new partition you must execute the stored procedures as described in ”Adding Partitions Using Stored Procedures”.

---

### Table 8–1  Schema Parameters Required by Oracle Scripter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db admin username</td>
<td>The database administrator's username.</td>
</tr>
<tr>
<td>db admin password</td>
<td>The database administrator's password.</td>
</tr>
<tr>
<td>db connection string</td>
<td>The database connect identifier. Refer to the Oracle Net Services documentation for the connect identifier syntax. Some examples include: ORCL localhost:1521/osm.us.example.com</td>
</tr>
<tr>
<td>ordermgmt owner schema</td>
<td>The core database schema name.</td>
</tr>
<tr>
<td>ordermgmt owner password</td>
<td>The core database schema password.</td>
</tr>
</tbody>
</table>

### Adding Partitions Using Stored Procedures

The *new_part.sql* script that is executed by Oracle Scripter runs the **ADD_HIST_PARTITION** and **REBUILD_IND_PARTITION** stored procedures in the **OM_PART_CONFIG_PKG** package. These procedures prepare SQL statements in the **OM_SQL_LOG** table, but do not run them. The script runs the **OM_PART_CONFIG_PKG.EXEC** procedure to perform those statements.

You may prefer to run the **ADD_HIST_PARTITION** and **REBUILD_IND_PARTITIONS** procedures directly instead of through *new_part.sql* script. For example, if you want to specify the tablespace for the new partition or review the prepared SQL statements:

1. Run SQL*Plus and connect to the schema.
2. Run the **OM_PART_CONFIG_PKG.INIT** procedure. This initializes the OSM tables used for preparing SQL statements.
   ```sql
   begin om_part_config_pkg.init(true);
   end;
   / ```
3. Run the OM_PART_CONFIG_PKG.ADD_HIST_PARTITION procedure, which prepares the SQL statements allocating new partitions. You can optionally specify the tablespace of the new partition as the input argument. If you do not supply it or the input argument is null, the partition is created on the tablespace of the table:

```sql
begin om_part_config_pkg.add_hist_partition;
end;
/
```

4. Optional: Review the prepared SQL statements:

```sql
select * from om_sql_log order by line;
```

5. Run the OM_PART_CONFIG_PKG.EXEC procedure, which processes the prepared SQL statements to add new partitions:

```sql
begin om_part_config_pkg.exec;
end;
/
```

6. Run the OM_PART_CONFIG_PKG.REBUILD_IND_PARTITIONS procedure.

This prepares SQL statements for rebuilding unusable indexes on the affected tables. In general, adding partitions does not render indexes unusable. However, it is safer to run this procedure.

```sql
begin om_part_config_pkg.rebuild_ind_partitions;
end;
/
```

7. Optional: Review the prepared SQL statements.

If you find no unusable indexes, the following query returns no rows:

```sql
select * from om_sql_log order by line;
```

8. Run the OM_PART_CONFIG_PKG.EXEC procedure, which processes the prepared SQL statements to rebuild any unusable indexes:

```sql
begin om_part_config_pkg.exec;
end;
/
```

Creating Partitions Automatically

When a block of order IDs is exhausted, the next block is allocated. If the corresponding partition does not exist, OSM will attempt to create it automatically. This is useful in development and testing environments. However, it is recommended that you add partitions manually in production environments, especially if you use Oracle RAC.

Adding partitions automatically during heavy processing can lead to order creation failures due to transaction timeouts, especially if your database storage is slow or orders are received over HTTP(S). Note that failed transactions initiated through JMS messages are rolled back and retried automatically when the JMS messages are redelivered.

For performance reasons, adding a new partition also involves a SPLIT PARTITION operation on om_order_flow_coordinator. If the database is an Oracle RAC database, this is needed to eliminate cluster waits. In addition, this improves performance when you drop old partitions because the script can drop entire om_order_flow_coordinator partitions, which is faster than row deletion.
The SPLIT PARTITION operation is performed on the om_order_flow_coordinator table after all ADD PARTITION operations succeed. If the system is busy processing orders, SPLIT PARTITION usually fails due to "resource busy" timeouts.

**Note:** The SPLIT PARTITION operation may create a partition with a size limit that is less than zero. The syntax for the name of the negative partition is N_ followed by a string of zeros up to the negative value. The string is always 18 character long.

For example, if the negative value were 100000, then the partition name is N_000000000000100000.

### Dropping Partitions with Oracle Scripter

You can drop existing partitions from a database schema if they have no pending orders, and you no longer require the legacy order data they contain.

To drop a partition, on the Windows machine where you installed the Database Utilities OSM Component, open the OSM Home\Database\osm\maintenance\dropPartitions\drop_partitions.sql script using Oracle Scripter and update the values of the parameters in Table 8–2.

**Table 8–2 Schema Parameters Required by Oracle Scripter**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>db admin username</td>
<td>The database administrator's username.</td>
</tr>
<tr>
<td>db admin password</td>
<td>The database administrator's password.</td>
</tr>
<tr>
<td>db connection string</td>
<td>The database connect identifier. Refer to the Oracle Net Services documentation for the connect identifier syntax. Some examples include:</td>
</tr>
<tr>
<td></td>
<td>- ORCL</td>
</tr>
<tr>
<td></td>
<td>- localhost:1521/osm.us.example.com</td>
</tr>
<tr>
<td>ordermgmt owner schema</td>
<td>The core database schema name.</td>
</tr>
<tr>
<td>ordermgmt owner password</td>
<td>The core database schema password.</td>
</tr>
<tr>
<td>max_order_limit</td>
<td>The maximum order limit.</td>
</tr>
</tbody>
</table>

After you save the updated parameter values, run the script using Oracle SQL*Plus.

The max_order_limit parameter places an upper limit on the set of orders to be deleted. Orders with ORDER_SEQ_ID greater than this limit are not deleted. Consequently, only partitions with an upper boundary less than or equal to max_order_limit + 1 are eligible for removal. A range partition is created with an upper boundary that is strictly less than a given value.

The script runs the drop_partitions stored procedure in the OM_PART_MAINTAIN package. It also handles database jobs, rebuilds any unusable indexes on the affected tables (global indexes become unusable when partitions are removed), and updates schema statistics.

The following issues apply when running the drop_partitions.sql script:

- Partitions with pending orders are not dropped.
- Partitions cannot be dropped if the schema contains only a single partition.
The script assumes that partition names follow the OSM naming convention for partitions (see the `partition_name` function in the `om_part_config_pkg` package).

As an example, consider an OSM database with partitions `P_000000000001000001`, `P_000000000002000001`, …, `P_000000000009000001`, and assume that all partitions but `P_000000000001000001` and `P_000000000003000001` have pending orders. Running `drop_partitions.sql` with 3000000 as `max_order_limit` drops only `P_000000000001000001` and `P_000000000003000001`.
This chapter helps you understand how Oracle Communications Order and Service Management (OSM) is related to WebLogic Server and the Oracle Database backup and restore procedures. For more information about these procedures, including warm or partial backup procedures, consult the Oracle Database and Oracle WebLogic Server documentation.

About Backing Up and Restoring OSM Files and Data

It is critical that you have a schedule and procedures for backing up and restoring your production-ready OSM system. This chapter includes a suggested schedule and backup and restore considerations, as well as information about the components involved in the backup. You must consider your own business needs when determining your backup and restore strategy.

Online backup of OSM data is supported if JMS JDBC store is configured.

Backup and Restore Overview

There are three main components to consider for backing up and restoring OSM. They can be performed in any order.

- Back up the OSM home directory (including the SDK directory and installation logs)
- Back up the OSM information from the Oracle Database
- Back up the files for the WebLogic domain

Backup and Restore Schedule

The three components of the OSM installation to be backed up require different schedules because they are modified at different times and for different reasons.

OSM Home Directory

The OSM program files in the OSM home directory do not change as part of the ongoing operation of OSM, so regularly scheduled backups may not be required. You should determine when it is appropriate to back up these files based on when you make changes to the files.

Oracle Database

You should perform a complete backup of the OSM database after installation. The suggested schedule for post-install backups is to take an incremental (level 0 in
RMAN) backup of the database monthly, a cumulative (level 1 in RMAN) backup weekly and a differential (level 1 in RMAN) backup daily. It is currently not possible to take consistent backups of database and transaction logs, because the transaction logs are file-based. For highest reliability use a highly available fault-tolerant storage (for example, SAN) for database and transaction log file stores.

WebLogic Server Files
Assuming that JMS JDBC store is configured, backups of the WebLogic domain directory and any external deployments should be done primarily after making changes to the configuration. These changes include adding deployments, changing domain configuration, and other administrative tasks.

If the persistent store for JMS is on the file system, that location should be backed up on the same schedule as the database.

The OSM attachments directory (see "OSM Attachment Directory") should also be backed up on the same schedule as the database, because it contains order-related data.

Backup and Restore Considerations
Overall considerations for OSM backup and restore include:

■ The backup of the WebLogic domain described in this chapter backs up not only the OSM application, but also the WebLogic configuration of any other applications that share the WebLogic domain with OSM. Restoring the domain will also return any other applications in the domain to the same restore point. Therefore, it is important (although outside the scope of this document) to coordinate the backup and restore of any applications that share the WebLogic domain.

■ Test backup and restore procedures in a test or staging environment before they are used in production.

Backing Up and Restoring the OSM Files
This section describes how to backup and restore the OSM files in the OSM_home directory.

Backing Up the OSM Files
The OSM installer creates files in a user-specified location during OSM installation. The default location in the installer is /opt/OSM, but usually another value is supplied during installation. This location is usually referred to as OSM_home.

You can back up these files using the tar command to put them in a single file, and then storing the .tar file in a safe location.

Restoring the OSM Files
To restore the OSM files, remove the contents of the OSM_home directory, and extract the contents of the backed-up tar file into the OSM_home directory.

Oracle Database Backup Considerations
The Oracle Database Server provides several means of backing up information. The two recommended methods for ordinary backup and restore are provided in this
section. There are no special considerations for OSM in determining the actual procedure for a backup or restore. Information about how to use the backup and restore methods considered in this section can be found in the Oracle Database documentation.

Database backup and restore procedures should be performed by a qualified database administrator.

RMAN Considerations

Recovery Manager (RMAN) is an Oracle Database utility that backs up, restores, and recovers Oracle databases. It backs up individual datafiles, and provides complete and incremental backup options. Following are some issues you should consider for using RMAN:

- Because it backs up datafiles, this method is most appropriate for use when OSM is not sharing any tablespaces with other applications. If OSM is sharing its tablespaces with other applications, they will be backed up at the same time. This means that if the OSM data is restored, the information for any other applications will be restored as well. This may not be desired.

- You should back up all of the permanent tablespaces that you have defined for OSM. For example, if you have different tablespaces for data and indexes, you should remember to back up both of them.

- RMAN may be slower than Flashback. This might be an issue in a large production environment.

Oracle Flashback Technology Considerations

Oracle Flashback Technology comprises a group of features that support the viewing of past states of data without needing to restore backups. It provides the ability to restore an entire database or individual tables from a set point in time. Following are some issues you should consider if you choose to employ this backup method:

- Because it backs up the entire database, this method is most appropriate for use when OSM is not sharing the database with other applications. If OSM is sharing the database instance with other applications, this method does not allow you to restore only the OSM portion of the database. This can cause data for other applications to be overwritten with older data.

- The Flashback Database command does not restore or perform media recovery on files, so you cannot use it to correct media failures such as disk crashes.

- Some editions of the Oracle Database software may not include this feature.

Backing Up and Restoring the WebLogic Server Configuration

The procedure in this section describes how to perform a full backup and restore of the WebLogic configuration for the domain used by OSM.

You must read through this entire section before starting the procedures. There may be pieces of information that you must retrieve from the domain before you shut it down.

There are several parts of the WebLogic Server configuration that should be backed up. They are not all required to be backed up on the same schedule, so consider the following when determining your WebLogic backup schedule. See "WebLogic Server Files" for more information about the backup schedule. The parts of the WebLogic configuration that must be backed up are the following:
Backing Up and Restoring the WebLogic Server Configuration

- WebLogic domain directory
- WebLogic persistent store
- OSM attachments directory
- External deployments
- Remote managed server directories (clustered server only)

**Backing Up the WebLogic Server Configuration**

This section relates to backing up WebLogic Server files for OSM.

**Before You Back Up the WebLogic Server**

There are some tasks that you must perform before starting the backup procedures in this chapter.

- Configure WebLogic so that when the domain is restarted, OSM will not process any JMS messages. This ensures that when you restore the domain from the backup, you can verify the configuration before OSM starts processing messages.
- Shut down all of the servers in the WebLogic domain that contains OSM. This includes any remote servers in a clustered environment.

**Setting OSM to Pause Processing of JMS Messages**

You can configure OSM to pause processing of JMS messages using the WebLogic Server Administration Console. By default, OSM has one JMS server, called `oms_jms_server`. If you have custom JMS servers defined, you should perform this procedure for each of them as well.

1. Access the WebLogic Server Administration console Home window. (See "Accessing the WebLogic Server Administration Console")
2. In the Messaging subsection of the Services section, click **JMS Servers**. The Summary of JMS Servers window is displayed.
3. Click on the name of the appropriate JMS server in the table.
4. The settings for your JMS server are displayed with the Configuration tab and the General sub-tab selected. At the bottom of the window, expand the **Advanced** heading to display more options.
5. Select the following options:
   - Insertion Paused At Startup
   - Production Paused At Startup
   - ConsumptionPaused At Startup
6. Click **Save**.
7. Exit the WebLogic console.

---

**Note:** After you have taken the backup and restarted the WebLogic Server, to return the server to its normal state by following the procedure above, but deselecting the options listed above.
WebLogic Server Domain Directory
The WebLogic Server domain directory contains many important parts of the server configuration, such as the main configuration files, security files, and LDAP files. You can back up these files using the `tar` command to put them in a single file, and then storing the `.tar` file in a safe location.

WebLogic Persistent Store
The default location for the persistent store for WebLogic is in a subdirectory of the domain directory. However, the persistent store may be required to be backed up separately, because it may be backed up on a separate schedule from the rest of the domain configuration. In addition, it is possible to configure this directory in a location outside the WebLogic domain directory. You can view the location of this directory in the WebLogic Server Administration Console. (For information about accessing the WebLogic Console, see “Accessing the WebLogic Server Administration Console”.) From the Home window of the WebLogic Console, Click Persistent Store in the right pane under Services. The Persistent Stores window is displayed containing a list of the persistent stores that have been defined and their types. Click the names of any file stores with a type of FileStore. The resulting window displays the location of the directory.

**Note:** If the Persistent Store type is listed as JDBCStore, you do not need to back it up separately, because it is backed up automatically with the database.

You can back up these files using the `tar` command to put them in a single file, and then storing the `.tar` file in a safe location. Ensure you back up the directories for all FileStore persistent stores if there are more than one.

OSM Attachment Directory
The default location for the attachments directory for OSM is in a subdirectory of the domain directory backed up in the previous section. However it is possible to configure this directory in another location. You can view the location of this directory in the WebLogic Server Administration Console. (For information about accessing the WebLogic Console, see “Accessing the WebLogic Server Administration Console”.) From the Home window of the WebLogic Console, Click FileT3 in the right pane under Services. The resulting window displays a list of the File (T3) Services that have been defined and their paths. If the path is not a complete path (for example, if it does not start with “/” on a UNIX system), the directory is located in the indicated location inside the domain directory.

You can back up these files using the `tar` command to put them in a single file, and then storing the `.tar` file in a safe location.

External Deployments
Any objects that are deployed into WebLogic using the OSM deployment tools are located in a subdirectory of the domain directory. If any objects have been deployed using other methods from a location outside the domain directory, these locations must be backed up as well.

You can use the WebLogic console to see if a particular object has been deployed from a directory outside the domain directory. (For information about accessing the WebLogic Console, see “Accessing the WebLogic Server Administration Console”.)
From the Home window of the WebLogic Console, Click **Deployments** in the right pane under Your Deployed Resources. The resulting window displays a list of the deployments in your domain. Click on the name of any deployment to see the path to the source file.

You can back up these files by copying them to a safe location.

**Remote Managed Servers (Clustered Server Only)**

If you are working in a clustered WebLogic environment, you must also back up the directories for any remote managed servers.

You can back up these files using the `tar` command to put them in a single file, and then storing the `.tar` file in a safe location.

---

**Restoring the WebLogic Server Configuration**

This section relates to restoring WebLogic Server files for OSM.

**Restoring the WebLogic Server Files**

To restore the WebLogic Server files for the following components, remove the contents of the directories you backed up, and extract the contents of the backed-up tar files into those directories.

- WebLogic domain directory
- WebLogic persistent store
- OSM attachments directory
- Remote managed server directories (clustered server only)

For the external deployments which you have backed up, copy the backed up files to the directory in which they were originally located, and replacing the existing files with the same names.

**Setting OSM to Resume Processing of JMS Messages**

Once you have restarted the WebLogic servers from the restored files and determined that the restoration has been successful, configure OSM to resume processing of JMS messages using the WebLogic Server Administration Console. By default, OSM has one JMS server, called `oms_jms_server`. If you have custom JMS servers defined, you should perform this procedure for each of them as well.

1. Access the WebLogic Server Administration console Home window. (See "Accessing the WebLogic Server Administration Console")

2. In the Messaging subsection of the Services section, click **JMS Servers**.

   The Summary of JMS Servers window is displayed.

3. Click on the name of the appropriate JMS server in the table.

4. The settings for your JMS server are displayed with the Configuration tab and the General sub-tab selected. At the bottom of the window, expand the Advanced heading to display more options.

5. Deselect the following options:
   - Insertion Paused At Startup
   - Production Paused At Startup
   - Consumption Paused At Startup
6. Click **Save**.
7. Exit the WebLogic console.
This chapter describes how to monitor and manage the Oracle Communications Order and Service Management (OSM) system using the Oracle WebLogic Server Console.

About Monitoring and Managing OSM

Oracle WebLogic Server requires a set of interrelated resources, such as database connections, Java Messaging Service (JMS) queues, execution threads, transactions, and system memory to work together in order to provide the functionality required of OSM.

You use the WebLogic Server Console to manage these resources, including tasks such as starting and stopping servers, balancing the load on servers or connection pools, selecting and monitoring the configuration of resources, detecting and correcting problems, monitoring and evaluating system performance, and making sure that OSM is correctly deployed to the target servers.

The WebLogic Server Console is a Web-based application that allows system administrators, support staff, and others to monitor and manage the OSM application remotely.

See the Oracle WebLogic documentation for more information.

Accessing the WebLogic Server Administration Console

To access the Oracle WebLogic Server Administration console do one of the following:

- If you are not connecting via Secure Socket Layer (SSL), enter the following URL into your browser:

  http://Hostname:Port/console

  where Hostname represents the DNS name or IP address of the computer on which the Administration server is installed and Port represents the address of the port on which the Administration server is listening for requests.

- If you are connecting via SSL, enter the following URL into your browser:

  https://Hostname:SSLPort/console

  where Hostname represents the DNS name or IP address of the computer on which the Administration server is installed and SSLPort represents the address of the port on which the Administration server is listening for SSL requests. This is a different port from the one used for non-SSL requests.
When started, the WebLogic Server Console prompts for a password. This should be the password for a user that is a member of the Administrators group in WebLogic. One such user is the WebLogic administration user that was created when the domain was created. By default, the name of this user is weblogic.

After you have successfully logged in, the WebLogic console Home window is displayed.

Using the WebLogic Console to Determine the Status of the OSM Application

After you have logged into the WebLogic console, you can access information about the status of the WebLogic servers and OSM deployments.

To access the status of the OSM server and deployments:

1. If you are not in the Home window of the WebLogic Console, click the Home icon in the upper left part of the window.
2. In the Environment subsection of the Domain Configurations section, click Servers.
   The Summary of Servers window is displayed. Server statuses are contained in the Health column of the table.
3. Click the Home icon in the upper left part of the window.
4. In the Your Deployed Resources subsection of the Domain Configurations section, click Deployments.
   The Summary of Deployments window is displayed. Deployment statuses are contained in the Health column of the table.

Note: If any of the deployments are not in the status that you expected, you can use the buttons on this window to start and stop individual deployments if necessary.

Managing Log Files

The following section details how to manage log files using the WebLogic Server Console.

Reading Log Files

You can view the OSM log files by opening the WebLogic Server Console.

To view the logs:

1. In the left pane of the console, click Diagnostics/Log Files.
2. From the Log Files table in the right pane, choose ServerLog, and then click View.
   The resulting page displays the latest entries in the log file. To view a specific, click to select the row, and then click View.
Configuring the Log View

OSM logs everything that happens in the system, which can make the logs file quite large. Using the WebLogic Server Console, you can configure the log views to show only the most recent messages.

To configure the view of the logs:
1. Log in to WebLogic Server Console.
2. From the log files page, click Customize this table.
3. Choose the appropriate filter and view options, then click Apply. To add an option to the Chosen column, highlight it in the Available column, then click the arrow that points to the Chosen column. To deselect the option, reverse this action.

Log Size and Rotation

It is important to be able to manage your log file maximum size and rotations to prevent the logs from filling up your disk resources.

To manage your log settings:
1. Log in to WebLogic Server Console.
2. Click Environment/Servers/server.
   This displays the General Configuration page for the selected server.
3. Click the Logging tab.
   The resulting page displays the logging settings for the selected server.
4. Modify the Rotation file size to 25,000 kilobytes and be sure to limit the number of Files to retain. Modify these settings for both tabs, General and HTTP.

Managing Error Message Volume and Logging Levels

OSM uses Log4j to generate and manage the system log messages. Through the use of Log4j, you can develop and maintain a logging strategy that minimizes the overall impact of logging operations on the application's resources. It does this by letting you control the volume of log messages generated.

You also have the ability, when necessary, to dynamically change, temporarily, the level and detail of the logged messages. This feature helps you to, for example, increase the level and detail of logged messages to help analyze performance problems within a production environment.

To read more about Log4j, refer to the Apache web site a:

http://logging.apache.org/log4j

Severity Levels

To control the volume of log messages generated and written to the output destinations (the console and the WebLogic log file, which are referred to as Appenders by Log4j), you assign severity levels to the various areas of the application that generate their own discrete messages (these areas are referred to as Categories by Log4).

The following is an ascending list of the severity levels, starting with the least severe:

- **Debug**: (least severe) designates fine-grained informational events that are most useful to debug an application
- **Info**: designates informational messages that highlight the progress of the application at coarse-grained level
- **Warning**: designates potentially harmful situations
- **Error**: designates error events that might still allow the application to continue running
- **Fatal**: (most severe) designates very severe error events that presumably lead the application to abort

If an event occurs inside a given category that triggers a message below the severity level assigned to the category (that is, less severe than the assigned level), Log4j does not generate the message.

**Example 1**
If you assigned the severity level **Warning** to a given category, Log4j does not generate any messages for that category that are flagged in the OSM code as **Debug** or **Info** level messages. Log4j will, however, generate all messages that are flagged as **Warning**, **Error**, and **Fatal**.

You can further control the number of messages written to the output destinations, or appenders, by also assigning a severity level to them. When you assign a severity level to an appender, it rejects messages below that severity level, even if Log4j passes the message to it.

**Example 2**
If you configure the console to the **Warning** severity level, and one of the categories is configured with the **Info** level, the console will not display the **Info** message, even though Log4j generates the message and passes it on to the console, because the message's severity level falls below the threshold for which the appender is configured. If, however, that same category later generates an **Error** level message, the console accepts and displays the message, because it carries a severity level equal to or higher than the console's threshold.

By default, the console and the WebLogic log file accepts all error messages, from the least severe to the most severe.

**Displaying Log Levels**
To display log levels:

1. Log in to WebLogic Server Console.
2. In the left pane of the Console, expand Diagnostics and select **Log Files**.
3. In the Log Files table, select the radio button next to the name of the log you want to view, and then click **View**.

   The page displays the latest contents of the log file; up to 500 messages in reverse chronological order. The messages at the top of the window are the most recent messages that the server has generated.

   The log viewer does not display messages that have been rotated into archive log files.

   The page displays the log file entry.

**Configuring the Severity Levels**
There are two methods by which you can configure the severity levels:
Managing Log Files

- **log4j.xml**: The log4j.xml file resides within the omslogging.jar of the OMS.ear file. Using this method you can change the default severity settings. This requires you to stop, then restart the servers to take effect. See "Configuring the log4j.xml File", below.

- **log4jAdmin**: The log4jAdmin web page lets you change the severity levels while the servers are running. The changes take effect immediately (this method is temporary; the system reverts to the default levels you established in the log4j.xml file when you restart the server). See "Configuring Log Levels Temporarily".

---

**Note:** To use this method, your Web Client log-in ID must have OMS_log_manager permissions.

### Configuring the log4j.xml File

Use this method to set up the default severity levels. To change these levels later, you must stop the server, modify the log4j.xml file, then restart the server.

If you need to modify the logging levels, you should use the log4jAdmin web page, as described in "Configuring Log Levels Temporarily".

There are two sections of the log4j.xml file that you should look at when configuring this file:

- **Appenders**: This section defines the output destination for the messages. At installation, the Appenders section contains two entries, as follows:
  - **Console**: From this entry you control the level of messages that the WebLogic console accepts.
  - **WebLogic**: From this entry you control the level of messages that the WebLogic log file accepts.

- **Categories**: This section contains references to all of the OSM categories that generate messages and gives you the ability to control the level of messages they generate.

To configure the log4j.xml file:

1. Unpack the oms.ear file. Copy oms.ear from domain_home/bin on the server to OSM_home\SDK\Customization.
2. Configure the unpackOMS.bat file, also located in OSM_home\SDK\Customization, with the correct location of JAVA_HOME for your environment.
3. Run the unpackOMS.bat file. This creates the osm-ejb\logging folder, where you will find the log4j.xml file.
4. Use a text editor to open log4j.xml.
5. Go to the **Console** entry in the Appenders section. To do this, search for the following string, which is at the top of the Appenders section:
   ```xml
   <!-- Append messages to the console -->
   ```
   The Console entry governs what level of messages are written to the console.
6. If necessary, change the threshold level. By default, it is configured to DEBUG, allowing the console to display all messages sent to it. If you want to restrict the number of messages displayed, change the threshold entry to the severity level...
Managing Log Files

appropriate for your installation (see "Severity Levels" above, for a description of the severity levels).

In the example below, the level is changed from DEBUG to INFO.

Before
<param name="Threshold" value="DEBUG"/>

After
<param name="Threshold" value="INFO"/>

7. Go to the WebLogic’s Log File entry in the Appenders section. To do this, search for the following:

<!-- Append messages to the weblogic's log file-->

The weblogic log file entry governs what level of messages are written to the WebLogic log file.

8. If necessary, change the threshold level as described in step 6.

9. Go to the Categories section. To do this, go to the top of the file and search for the string:

   category name

This takes you to the first category entry in the file.

10. Review each of the categories in this section, changing the severity level where necessary.

    In the example below, the level is changed from INFO to WARNING.

    Before
    <category name="org.jboss.system">
    <priority value="info"/>
    </category>

    After
    <category name="org.jboss.system">
    <priority value="warning"/>
    </category>

11. When you finish updating the categories, save and close the log4j.xml file.

12. Repack the oms.ear file. At the command prompt, run packOMS.bat.

13. Re-deploy the oms.ear file.

Configuring Log Levels Temporarily

Use log4jAdmin web page to check the current logging levels or to change the logging levels dynamically.

---

Note: You can use this method only to change the severity levels of the Categories. To change the Appender level (the logfile output, for example, console or file), you must reconfigure the log4j.xml file (see "Configuring the log4j.xml File" for an explanation of the Categories, the Appenders, and how to configure the log4j.xml file).
The changes you make to the logging severity levels using this method are temporary; they are not written to the log4j.xml file. When you restart the server, the logging levels return to those that are configured in the log4j.xml file.

Note: To perform the following procedures, your log-in ID must belong to the OMS_log_manager group. This set of permissions can be granted through the WebLogic Server console.

Checking the current logging levels

You can use the Filter Loggers feature to check the logging level of specific categories or sub components. If you know the name of the category or sub-component that you want to check, you can use the filter to display only that category, or related, categories.

To check the current logging level:

1. To open the log4jAdmin web page, enter the following path in the browser's address line (Note: the URL is case sensitive):
   
   http://localhost:7001/OrderManagement/control/log4jAdmin

2. Enter the beginning of the name, or a part of the name, in the Filter Loggers field

3. Click either:
   - Begins With (if filtering on the beginning of the name)
   - Contains (if filtering on just a part-name)

4. The list displays the categories and subcategories that match the entry in the Filter Loggers field.

Note: To change the actual logging level from the list that you get from the Filter Loggers, use the following procedure.

To change the logging levels dynamically:

1. Open the log4jAdmin web page, enter the following path in the browser's address line (Note: the URL is case sensitive):

   http://<host:port>/OrderManagement/admin/log4jAdmin

2. Scan down the entries in the left-hand column of the page. This column contains a list of the categories and their related sub-components. Find the category or sub-component for which you want to change the logging level.

   Example

   com.mslv.oms.automation (category name)

   or

   com.mslv.oms.automation.AutomationDispatcher (sub-component name)
3. Scan across the row to the severity levels. The level that currently is selected is highlighted in a different color from the other levels and appears in the Effective Level column.

4. Click the level to which you want to make the change. To:

   - Change an entire Category: Click the category name
   - Change just the sub-component: Click the sub-component name

   The change takes place immediately.

   **Note:** If you know the name of the category or sub-component that you want to change, you can use the filter at the top of the page to display just that, or related, categories. Enter the beginning of the name, or a part of the name, in the Filter Loggers: field, then click Begins With (if filtering on the beginning of the name) or Contains (if just filtering on a part-name).

   To show specific category and sub-component names, at least one corresponding order should have been previously submitted to OSM. For example, to show sub-components beginning with "oracle.communications.ordermanagement.requestprocessor", at least one instance of the corresponding SalesOrder10000DeliverEBM.xml sample order should have been previously submitted.

5. Repeat step 4 for as many categories or sub-components that you need to change.

6. When you finish making the changes, close the page.

---

### Monitoring Performance

The WebLogic Server Console provides a real-time view of system performance.

You can access the performance monitor by using the following procedure.

To access the performance monitor:

1. Start the WebLogic Server Console.

2. Click **Environment/Servers/server**. This displays the General Configuration page for the selected server.

3. Click **Monitoring**.

4. Click the **Health** tab to view the health status for all OSM related sub-systems. If the status is not **OK**, review the reason and, if required, access the server log for more information.

   Health status severity levels are shown in the bottom left pane of the console under System Status. (OK, Warn, Overloaded, Critical, Failed)

5. Click the **Performance** tab to view JVM memory utilization statistics for the server.

   If the memory usage statistics are high, you must allocate more memory to the Java runtime by increasing the `-Xms` and `-Xmx` parameter values. The file that
contains this parameter depends on the operating system you are using and the WebLogic Server to which the OSM server is deployed. An example is shown below.

Example:

Open the file `${DOMAIN_HOME}/bin/setDomainEnv.sh` and provide the following parameter values:

```
MEM_ARGS="-Xms2048m -Xmx2048m"
```

The more memory that you allow for the Java runtime, the faster OSM will run under high loads.

6. Click the Threads tab to monitor thread activity for the server. Important columns to monitor are Queue Length and Pending User Request Count. A count of zero is optimal, meaning no user requests are stuck or waiting to be processed.

If any of the counts are unusually high in the pool, go to the second table to troubleshoot the individual threads.

7. Click the Workload tab to monitor the Work Managers configured for the server. If the Pending Requests count is not zero, you should access the server log file for more information or perform a thread dump.

8. Click the JDBC tab to monitor the database pool connections configured for the server. The installer creates a maximum capacity of 15 connections for the connection pool.

If the Active Connections High Count is 15 and the Active Connections Average Count is half of that, you may need to increase the number of connections. See "Managing Database Connections" for more information.

9. Click the JTA tab to monitor transaction activity on the server. There should be no Roll Back statistics in the summary; if so, refer to the server log file for more information.

### Managing Database Connections

In OSM, database connections are managed through database pools that are set up in WebLogic.

The database pool connections can be configured through the WebLogic Server Console by clicking Services/JDBC/Data Sources/oms_pool/Connection Pool.

Use the parameters on the Connection Pool page to modify and fine tune your connection pool settings as the need arises.

### Using JMS Queues to Send Messages

OSM uses JMS Queues and Topics which are both JMS Destinations. Queues follow a point-to-point communication paradigm, while Topics follow the publish and subscribe paradigm.

**Note:** In an OSM clustered environment, you must use JMS queues as a JMS destination to receive JMS events. Do not use JMS topics in an OSM clustered environment.
When OSM sends data to an external system, such as UIM or ASAP, it does so by sending JMS messages to the appropriate JMS request queue of an external system.

If the external system is not processing the requests from OSM, the queues get backlogged. It is important to be able to monitor the size of the JMS queues in order to know whether or not they are backing up.

To monitor the JMS queues:

1. Login to WebLogic Administration Console
   Click Services/Messaging/JMS Servers/oms_jms_server.
   The General Configuration page is displayed.

2. Click the Monitoring tab then click Active Destinations.
   A list of active destinations targeted to the server is displayed.

   **Note:** The default view of the table does not contain the Consumers column. We recommend that you customize the table using Customize link to include this column, along with any other customizations you may want to make.

The Consumers column defines the current number of listeners on the destination. If a destination does not have any listeners, then the external system will not receive the messages.

The Messages Current column defines the current number of unprocessed messages in the JMS destination. A large number (for example, 10,000) in this destination is a problem. It means that the messages are not getting processed, or that the messages are getting processed but errors are occurring and the messages are getting put back on the destination.

When OSM is first installed, the following JMS destinations are present:
- oms_behavior_queue: Used for customizing task assignment
- oms_events: Internal destination used for events such as automation, notifications, and task state changes
- oms_order_events: Used for order state changes such as OrderCreateEvent, OrderStateChangeEvent, AmendmentStartedEvent, OrderCancelledEvent
- oms_order_updates: Internal destination used for processing amendments
- oms_signal_topic: Internal destination used to trigger a metadata refresh

### Monitoring the Event Queue

The destination oms_events is the JMS destination to which OSM events are sent. OSM events are sent when tasks change states, or when notifications occur.

The number of consumers for the oms_events is determined by which plug-ins are configured. If plug-ins are configured, the number of consumers must not be 0.

If there is a problem with automation plug-ins getting invoked, check the consumers queue and the messages queue.

If the consumers queue is less than the number of plug-ins, the plug-ins are not configured correctly. Check the OSM_home\SDK\Samples\DatabasePlugin\map\automationmap.xml file and make sure that all of the plug-ins have been deployed.
If the messages queue keeps getting larger, the plug-ins may not be committing the transactions during processing of the events. Verify the plug-in code and check the log files.

**Sending Data to External Systems Using Plug-Ins**

If there are external systems deployed to the same WebLogic instance as OSM, when you monitor the JMS destinations, watch for the following.

---

**Note:** The important columns are Consumers, Messages, and Messages Received.

---

If the number in the messages column for these queues continues to grow, the external system may not be processing the messages sent by OSM. You must check to see if the external system is working properly.

If the number of consumers for the queues is 0, such as UIMrequestQueue, the external system may not have configured its listeners properly. Check to see if the external system is configured properly.

**Updating the JMS Redelivery Configuration Settings**

When the Order-to-Activate cartridges are installed, the Redelivery Delay Override and Redelivery Limit WebLogic parameters are set during installation to 7000 msec and 10, respectively. However, different values may be more effective for your OSM environment depending on your usage of the system.

If you encounter timing related issues for message delivery on JMS queues, there are a number of WebLogic settings that you can modify to resolve the issue. These values are set on every JMS queue through the WebLogic Service Console. From Home, select JMS Modules, and then select oms_jms_module to modify the following settings:

- **Redelivery Delay Override:** Delay in milliseconds before rolled back or recovered messages are redelivered. This value overrides the Redelivery Delay setting.
- **Redelivery Limit:** The number of times to attempt to redeliver a message.
- **Time-to-Deliver:** Delay in milliseconds before a sent message is visible at the target destination. Typical values for this setting are 100 through 700.

To find the best values for these parameters, start with initial values less than 7000 msec for the Redelivery Delay Override, 10 for the Redelivery Limit and 100 for the Time-to-Deliver parameter and increase them slightly until no occurrences of errors are observed. The actual values you finalize on will depend on your particular implementation of OSM. See the Oracle WebLogic documentation for complete details on these parameters.

**About OSM and XA Support**

The OSM database does not support XA transactions because the Oracle thin-client driver used for JDBC connections does not support XA. However, the OSM WebLogic Server configuration uses an XA emulation feature in order to get a two-phase commit across JMS/JDBC automation transactions.

Even though OSM uses a non-XA driver for database transactions, external XA resources can still participate in transactions. For example, JMS bridges can be XA-enabled for an outside application, but the OSM side of the transaction will still
use the non-XA emulated two-phase commit. Note that this also applies to JMS queues which support Application Integration Architecture (AIA) cartridges.

**Overriding the Internet Explorer Language in the OMS Web Clients**

If the Internet Explorer installation that is used to access the OMS web clients is set to a language other than English, and this language matches one of the properties files included in the `oms.ear` file, the Web client prompts appear in the non-English language.

The language used in the Web clients is controlled by the `resources.properties` file. Additional language property files in the `oms.ear` file include:

- `resources_cs.properties` (Czech language properties file)
- `resources_zh.properties` (Chinese language properties file)

To remove support for a non-English language, unpack the `oms.ear` file, remove the corresponding properties file, repack the `oms.ear` file and redeploy it.

For example, if the browser is set to use the Czech language, and the `resources_cs.properties` file exists in the `oms.ear` file, the Web client prompts appear in Czech. Removing the `resources_cs.properties` file causes the Web client prompts to appear in English, even though the browser language setting is still configured to the Czech language.
Using the XML Import/Export Application

This chapter provides information about the XML Import/Export application (XMLIE), which is used to manage data and metadata in the Oracle Communications Order and Service Management (OSM) database schema.

About the XML Import/Export Application

XMLIE is included with the OSM SDK Tools component that can be installed with the OSM installer. For more information about installing the OSM SDK Tools component, see OSM Installation Guide.

There are two types of information in an OSM database schema:

- Metadata: Information that defines the order model. For example, the definitions of processes, orders, and tasks.
- Data: Information that represents orders. For example, order nodes, attributes, and values.

Using XMLIE, you can perform actions such as import and export metadata, purge metadata and data, and migrate data. You can also use XMLIE to validate the metadata model, and to create a graphical representation of the metadata.

**Note:** Although actions such as importing and exporting metadata, purging both metadata and data, and migrating data can be done using XMLIE, Oracle Communications Design Studio is the preferred application for executing these functions.

**Note:** XMLIE can work with a localized database, but the application must also be localized. See OSM Developer’s Guide for information on localizing OSM, including localizing XMLIE.

If you are running the OSM application on a UNIX or Linux platform, you must run XMLIE by using a set of Ant scripts. If you are running the OSM application on a Windows platform, you must run XMLIE by using a set of batch scripts.

About Using the XML Import/Export Application

The following steps provide a high-level overview of using XMLIE:

1. Configure the XMLIE environment files:
For Ant commands, configure the `OSM_home/SDK/XMLImportExport/build.properties` file (where `OSM_home` is the location of the base OSM installation directory). See “Configuring the build.properties File for Ant Commands”.

For batch scripts, configure the `OSM_home/SDK/XMLImportExport/config.bat` script. See “Configuring the config.bat Script for a Batch Scripts”.

2. Copy the `OSM_home/SDK/XMLImportExport/config/config_sample.xml` file and rename it to `OSM_home/SDK/XMLImportExport/config/config.xml` in the same directory.

   **Note:** OSM creates the `OSM_home/SDK/XMLImportExport/config/config_sample.xml` file when you install the OSM SDK Tools component. This file contains a sample XMLIE configuration file that can be used as a template for the `config.xml` configuration file.

   The `config.xml` file name is arbitrary. If you customize the name of the `config.xml` file, ensure that you substitute the customized name wherever you must specify the `config.xml` file (for example, when using the import and export commands in the `import.bat` and `export.bat` scripts).

   This chapter uses the default `config.xml` file name in all examples.

3. Configure the `OSM_home/SDK/XMLImportExport/config/config.xml` file.

   Both the XMLIE Ant commands and batch scripts use this file to define:

   - Connections to other components.
     For example, database connection XML node that provides the OSM schema user name, password, and connection details.
   - How the Ant commands and batch scripts work.
     For example, the `import.bat` script is configured using the `import` node in the `config.xml` file. This node contains elements that specify whether the imported data is validated, what actions to take if the database is not empty, and what actions to take if the XML model you are importing already exists in the database.
   - The data or metadata on which the commands are to act.
     For example, you can configure a selective import that only imports one specific element into an existing OSM cartridge using the `selection` element.

   See “Configuring the config.xml File XML Import/Export Nodes and Elements”.

4. You can create an empty text file with a `.xml` extension and then use the `export.bat` script or the `ant export` command to populate the file. Then you can edit it for use with other batch scripts or ant commands.

   **Note:** This chapter uses `xmlModelFile` as the documentation placeholder name for this file.

5. Run the ant command or batch script.
About XML Import/Export Batch Scripts and Ant Commands

The following sections describe the XMLIE batch scripts and ant commands.

About XML Import/Export Ant Commands and Syntax

OSM supports the following ant commands and syntax for UNIX and Linux systems:

- **ant export**: exports all cartridges, all cartridges within a given namespace, or one specific cartridge.
- **ant import**: imports the XML model.
- **ant purge**: completely purges the target DB.
- **ant undeploy**: undeploys the given cartridge if no pending orders exist.
- **ant forceUndeploy**: forces to undeploy the given cartridge including pending orders.
- **ant EncryptPasswords**: Encrypts passwords contained in the config.xml file. For more information, see "Using the EncryptPasswords Utility".
- **ant convert**: upgrades the old XML model to latest version.
- **ant refresh**: refreshes the metadata in the WebLogic Server.
- **ant htmlModel**: converts an XML metadata document to HTML and graphical format.
- **ant userAdmin**: adds users from WebLogic groups and OSM workgroups.

For more information about the **ant userAdmin** script, see "Using the XML Import/Export Application to Administer Users and Workgroups".

Before you can run these ant commands, you must configure the ant environment build.properties file and the config.xml file. For more information see "Configuring the XML Import/Export Environment Files" and "Configuring the config.xml File XML Import/Export Nodes and Elements".

About XML Import/Export Batch Scripts and Syntax

OSM supports the following batch scripts for Windows systems in the OSM_home/SDK/XMLImportExport folder:

- **import.bat**: Imports an XML metadata document into an OSM database.
- **export.bat**: Exports an OSM database to an XML metadata document.
- **migrate.bat**: Migrates order data from one version of a cartridge to another version of the same cartridge.
- **purge.bat**: Purges the entire OSM schema (metadata and orders) or undeploys a specific cartridge.
- **orderPurge.bat**: Purges orders that satisfy purge criteria; can be run on an immediate or scheduled basis.
- **validate.bat**: Validates an XML model document.
Configuring the XML Import/Export Environment Files

- **modeldoc.bat**: Converts an XML metadata document to HTML and graphical format.
- **EncryptPasswords.bat**: Encrypts passwords contained in the config.xml file. For more information, see "Using the EncryptPasswords Utility".
- **convertmodel.bat**: Migrates a model from older versions and uses the following syntax:

```
convertmodel.bat xmlModelFile
```

**Note**: This script is deprecated. Use the import.bat script instead because it automatically upgrades models during an import.

- **userAdmin.bat**: Adds users to WebLogic from WebLogic groups and OSM workgroups based on an XML definition.

For more information about the **userAdmin.bat** script, see "Using the XML Import/Export Application to Administer Users and Workgroups".

**Note**: The import.bat, export.bat, migrate.bat, purge.bat, and validate.bat scripts can use the -p db_password argument (where db_password is the password for the OSM schema) to provide the OSM schema password from the command line. In addition, the migrate.bat script can use the -clientpassword xmlAPI_password argument (where xmlAPI_password is a WebLogic password for a user performing the migration). These arguments have been deprecated. For security recommendations, see "Configuring the config.xml File XML Import/Export Nodes and Elements".

Before you can run these scripts, you must configure the environment **config.bat** script and the **config.xml** file. For more information see "Configuring the XML Import/Export Environment Files" and "Configuring the config.xml File XML Import/Export Nodes and Elements".

### Configuring the XML Import/Export Environment Files

The following sections describe the files you need to edit to configure the environment for the XMLIE ant commands and batch scripts.

### Configuring the build.properties File for Ant Commands

The paths to the **config.xml** file and the XML model document, along with cartridge and environment properties used by the XMLIE ant commands must be specified in the **OSM_home/SDK/XMLImportExport/build.properties** file. Unlike the XMLIE batch scripts, you do not have to specify these paths in the command line.

To configure the **build.properties** file for ant commands:

1. Update the following variables in the **OSM_home/SDK/XMLImportExport/build.properties** file:

```properties
java.maxmemory=java.maxmemory
xmlie.root.dir=root.dir
xmlie.root.modelDocument=modelDocument
```
xmlie.root.configDocument=configDocument
xmlie.root.namespace=namespace
xmlie.root.version=version
xmlie.root.htmlDir=htmlDir

where:

- **java.maxmemory**: The maximum heap to be used by JVM
- **root.dir**: The path of XMLIE directory
- **modelDocument**: The path for XML model document
- **configDocument**: The path for config.xml
- **namespace**: The OSM cartridge namespace
- **version**: The cartridge version
- **htmlDir**: The path for HTML model directory

**Note:** You can also refer to the README.txt file found under SDK/XMLImportExport for information on configuring the build.properties file for ant scripts.

For example:

```
java.maxmemory=512m
xmlie.root.dir=./
xmlie.root.modelDocument=./data.xml
xmlie.root.configDocument=./config/config.xml
xmlie.root.namespace=test
xmlie.root.version=4.0
xmlie.root.htmlDir=./htmlModel
```

2. If you want to configure order purge parameters in the build.properties file, see "Running Ant with the orderPurge.xml file On UNIX or Linux to Purge Orders".

3. If you want to configure cartridge migration parameters in the build.properties file, see "Configuring and Running an Order Migration".

**Configuring the config.bat Script for a Batch Scripts**

To configure the batch script environment:

1. Update the OSM_home\\SDK\\XMLImportExport\\config.bat script to specify the following paths and configuration options:

**Note:** No other configuration values in the config.bat script should be edited.

- **JAVA_HOME**: The directory in which the Java SDK or Java Runtime Environment resides.
- **APP_ROOT**: The XML handler root directory. If the XMLIE application is installed in the default location, you must enclose the path in double quotation marks, as in: "C:\Program Files\OSM\SDK\XMLImportExport".
- **JAVA_OPTS**: The minimum and maximum memory requirements.
For example:

```
set JAVA_HOME=C:\Oracle\Middleware\jdk1.6.0_32
set APP_ROOT="D:\osm\SDK\XMLImportExport"
set JAVAOPTS=-ms512m -mx512m
```

### Configuring the config.xml File XML Import/Export Nodes and Elements

To configure the `config.xml` file:

1. Open the `OSM_home/XMLImportExport/config/config_sample.xml` file.
2. Create a `config.xml` file by copying `config_sample.xml` to `config.xml`.

#### Note:
The `sample_config.xml` file contains references to absolute paths that start with "C: \". Be sure to configure these paths to reflect your environment.

3. Configure the database connection node in the `config.xml` file. This node is required for most ant commands and batch scripts.

   ```xml
   <databaseConnection>
     <user>osm_schema</user>
     <password>osm_schema_pwd</password>
     <dataSource>jdbc:oracle:thin:@ip_address:osm_database_port:osm_database_sid</dataSource>
   </databaseConnection>
   ``

   where:

   - `osm_schema` and `osm_schema_pwd` are the OSM schema user name and password.
   - `ip_address`, `port`, and `sid` are the OSM database IP address, port number, and SID.

   **Note:** To export / import the symbols change the character set and db connection to:

   ```xml
   <encoding>UTF-8</encoding>
   <dataSource>jdbc:oracle:oci: (description=(address=(host=kanlddb698.ca.oracle.com)(protocol=tcp)(port=1521)) (connect_data=(SID=ORCH10G)))</dataSource>
   ```

4. Configure the XML API connection node to specify the connection information to be used by operations that utilize the XML API. For example, migration ant commands or batch scripts require this node.

   ```xml
   <xmlAPIConnection>
     <user>weblogid_user</user>
     <password>weblogic_pwd</password>
     <url>http://ip_address:port</url>
   </xmlAPIConnection>
   ```

   where
- `weblogic_user` and `weblogic_pwd` are the user name and password of the user performing the migration. This user must have access to all source orders being migrated and to target order's type/source order entry task (if `closeSource` set to true, see "About Migrating Orders" for details).

- `ip_address`, and `port` are the WebLogic Administration server IP address and port number.

5. Configure the WebLogic Administrator credentials and connection information node. This node is required for ant commands and batch scripts that modify OSM user credentials:

```xml
<j2eeAdminConnection>
  <j2eeServiceName>weblogic</j2eeServiceName>
  <user>weblogid_user</user>
  <password>weblogic_pwd</password>
  <hostname>ip_address</hostname>
  <port>port</port>
</j2eeAdminConnection>
```

where

- `weblogic_user` and `weblogic_pwd` are the user name and password of the WebLogic with Administrator privileges.

- `ip_address`, and `port` are the WebLogic Administration server IP address and port number.

6. Configure the XMLIE log file location node:

```xml
<log logFileUrl="file:/:filename_path" overwrite="boolean"/>
```

where:

- `filename_path` is the path to the log file that includes the file name of the log file.

- `boolean` can be true or false. If set to true (default), XMLIE overwrites the log file every time the application starts. If set to false, XMLIE creates a cumulative log by saving the log from session to session and adding new messages to it.

7. Do one of the following:

- For importing metadata to the OSM database, configure the import node. See "About Importing Metadata".

- For exporting metadata from an OSM database, configure the export node. See "About Exporting Metadata".

- For purging metadata and order data in an OSM database, configure the purge node. See "About Purging MetaData and Data".

- For migrating orders between OSM cartridges, configure the migrate node. See "About Migrating Orders".

- For validating Metadata from an existing XML file, configure the validation node. See "About Validating the Metadata Model and Data".

- For creating a graphical HTML representation see "About Creating a Graphical Representation of the Metadata Model".

8. (Optional) If XMLIE is to be run unattended, secure the EncryptPassword utility and the configuration file that contains user credentials.
For enhanced security, each of the XMLIE operations that require user passwords prompts you for those passwords during invocation. If XMLIE is to be run unattended, you can alternatively encrypt those passwords and store them in the XMLIE configuration file (typically `config.xml`).

If passwords are to be stored in the XMLIE configuration file, do the following:

a. Set the permissions of the configuration file to be readable only by select administrative users. Refer to your OS documentation for instruction.

b. Run the EncryptPassword utility so that user name and password credentials for all XMLIE users are encrypted for safe storage. See "Using the EncryptPasswords Utility" for more information.

---

**Note:** If you plan to run XMLIE in an unattended mode, you must first run the EncryptPasswords utility; otherwise, you cannot perform many of the application functions and OSM gives an error indicating that you must run the EncryptPasswords utility.

---

### About Importing and Exporting Metadata

You can transfer metadata from one OSM database to another when setting up a new OSM environment. For example, you may want to set up multiple OSM environments of the same version, such as development, test, and production environments. Or, you may want to set up a new version of an OSM production environment based on a previous version of an OSM production environment.

Transferring metadata from one OSM database to another is a combination of an export from the one OSM database followed by an import to another. You can also import and export selected parts of the metadata, as opposed to all of the metadata.

---

**Note:** The actions that XMLIE runs are supported in Design Studio. Any metadata changes made using XMLIE will be overwritten when deploying the same cartridge using Design Studio.

---

The following sections describe import and export commands and configurations.

### About Exporting Metadata

XMLIE provides the `export.bat` script or the `export` ant command, which is used to export metadata from an OSM database. Exported metadata is stored in an XML file (the `xmlModelFile`). You can export metadata using Design Studio; however, this functionality remains available through XMLIE as well.

You can export the entire metadata database, or you can use the export `selection` element to specify the metadata to export based on:

- All entities or selected entities from a specific cartridge
- All cartridges in a namespace

Each OSM cartridge is uniquely identified by namespace and version, so exporting all of the cartridges in a namespace includes all versions of the cartridge within the namespace.
About the Order of Exported Metadata

When the order of the resulting list makes no logical or significant difference, XMLIE places the metadata files in ascending alphabetical order. This ensures that when you do repeated exports of the same database, the metadata files will always appear in the same order, which makes it easier to merge metadata changes with future development.

In Example 11–1, the definition of entities (for example, each state) is now sorted alphabetically by name:

Example 11–1 Definition of Entities (each state) Sorted Alphabetically by Name

```xml
<state name="accepted">  
<description>Accepted</description>  
</state>  
<state name="assigned">  
<description>Assigned</description>  
</state>  
<state name="completed">  
<description>Completed</description>  
</state>  
<state name="received">  
<description>Received</description>  
</state>  
...  
```

In Example 11–2, the references to entities (for example, each state) is now sorted alphabetically by name:

Example 11–2 References to Entities (states or statuses) Sorted Alphabetically by Name:

```xml
<task name="enter_payment_information" xsi:type="genericTaskType">  
<description>Enter Payment Information</description>  
<state>accepted</state>  
<state>completed</state>  
<state>received</state>  
<status>back</status>  
<status>next</status>  
</task>  
```

In Example 11–3, the Import/Export application does not sort by name because the order matters (that is, there is a logical difference). Country appears after last_name because the designer specified country to appear after last_name:

Example 11–3 Not Sorted By Name

```xml
<masterOrderTemplate>  
<dataNode element="account_information">  
<dataNode element="first_name"/>  
<dataNode element="last_name"/>  
<dataNode element="country"/>  
&viewRule xsi:type="eventRuleType">  
<event>value-changed</event>  
```
About Export Layout Options
When you perform a selective export, you can specify different export layouts by using the export command to create one or multiple files, and where to put the files. You have the following options:

- Use the `singleDocument` layout to export to a single file that contains all of the exported entities.
- Use the `cartridge` layout to export to one main document and a single folder for each exported cartridge. Each cartridge folder contains a version folder.

Figure 11–1 shows the `cartridge` layout.

Figure 11–1  Cartridge Layout

- You can use the `entity` layout to export to:
  - One main document
  - A folder for each cartridge included in the export (you can specify that the export include entities from more than one cartridge)
  - A folder for the cartridge version
  - A single folder for each entity type exported. Under the entity type folders are the individual entity XML files

Figure 11–2 shows the `entity` layout.
Keeping the ID Integrity in SQL Rules

Note: This section is applicable only if you are upgrading up to OSM 7.0 from a previous release. SQL rules and text rules were replaced in OSM 7.0. SQL rules are supported in Design Studio. The SQL rule is imported as a separate file that can be edited as a text document.

The SQL based rule type in OSM can contain entity IDs (mostly node IDs), which must be replaced with new IDs during data migration. IDs must be exposed as entity attributes, helping the application find them in SQL based rules, and replace them with new ones while importing them into a fresh environment. IDs in a document otherwise serve no purpose and should be ignored. You can set this feature to false by setting the exposeEntityID, in the config.xml file for XMLIE, to false, thereby reducing the model document size.

Text rules can reference any entity ID. Generally, however, node IDs used in these known patterns for other possible entity ID conversion routines require user assistance to tokenize the rule text for proper parsing and conversion. A token suggested before the ID must be tokenized for IDs except known patterns (all node functions and stored procedures in om_ordinst_value_pkg package):

/*$entityType*/

Example 11–4 Original Rule

declare val1 date;
 delay_flag    varchar2(10);
begin
 select timestamp_out into val1
 from om_hist$order_header hist, om_task task, om_state st
 where hist.task_id = task.task_id
 and node_id = /*$dataNode*/76983
 and hist.hist_order_state_id = st.state_id and st.state_mnemonic = 'completed'
Example 11–5 Modified Rule
The import operation detects IDs and replaces them with new IDs.

```
deflag := om_ordinist_value_pkg.get_node_value_like(:order_seq_id, new_id, :coord_set_id);
deflag := om_ordinist_value_pkg.get_node_value_like(:order_seq_id, 76983, :coord_set_id);
if ( rtrim(delay_flag)='yes' ) or (val1 <= (sysdate - 2/24)) then
  :rule_result := 'true';
else
  :rule_result := 'false';
end if;
end
```

Configuring and Running an Export
To configure and run an export:

1. Configure the `config.xml` file for use with the export ant command.

   ```xml
   <export validateModel="validation" exposeEntityID="entityID" layout="layout_action" readOnlyFileAction="readOnly_action"/>
   </export>
   
   where:

   ■ `validation`: Options are:
     - `true`: Validates the XML model before performing the export.
     - `false`: Does not validates the XML model before performing the export. (default).

   ■ `entityID`: Options are:
     - `true`: Exposes the EntityID in the exported file. (default)
     - `false`: Conceals the Entity ID in the exported file.

   **Caution:** Oracle recommends you do not skip the model validation, so this parameter should always be set to `true`.  

About Importing and Exporting Metadata

1. **layout_action**: Options are:
   - **singleDocument**: Exports everything to a single file. (default)
   - **cartridge**: Creates a single file for each cartridge exported.
   - **entity**: Creates a single file for each entity exported.

2. **readOnly_action**: Options are:
   - **ignore**: The Export operation will not overwrite read-only files. (default)
   - **replace**: The Export operation will overwrite read-only files.

   (Optional) Add the **selection** element within the **export** node to export targeted entities to OSM metadata using an XPath expression. Use the following syntax to define the selection element:

   ```xml
   <export validateModel="validation" exposeEntityID="entityID" layout="layout_action" readOnlyFileAction="readOnly_action">
     <selection>/oms:model/oms:cartridge[@namespace="namespace" and @version="version"]/oms:entity</selection>
     <selection>/oms:model/oms:entity</selection>
   </export>
   ``

   where:
   - **namespace**: The namespace for the cartridge.
   - **version**: The cartridge version.
   - **entity**: The entity you are targeting. For example, workgroup, region, and schedule.

3. Do the following:
   a. If you using ant, run the following command:
      ```none
      ant export
      ```
   b. If you using a batch script, run the following command:
      ```none
      export.bat xmlModelFile config/config.xml
      ```

About Importing Metadata

XMLIE provides the **import** command, which is used to import metadata into an OSM database. If you import metadata, make sure that the elements you import do not conflict with existing metadata that is part of a Design Studio OSM cartridge. Otherwise you may encounter version conflict, overwrite existing elements, and create other discrepancies.

You can import the entire metadata database using the import node or you can use the import **selection** element within an **import** node to specify the metadata to import based on:

- All entities or selected entities from a specific cartridge
- All cartridges in a namespace

Each OSM cartridge is uniquely identified by namespace and version, so exporting all of the cartridges in a namespace includes all versions of the cartridge within the namespace.

- System level parameters
By enabling selective imports, you can grant concurrent access to a single model or cartridge for multiple developers. Using this method, developers can import just the entities on which they are working at that moment, which gives other developers access to other entities within the cartridge.

The import operation is performed in one transaction. Consult your Oracle Database Administrator (DBA) for the appropriate setup for the rollback segment.

**Note:** User workgroups are not part of the metadata model, so they must be re-entered after an import.

**Note:** After importing or exporting a cartridge, you must remap the e-mail notifications that were associated with individual users. For best results, associate notifications only with workgroups because user names differ between environments.

### Configuring and Running an Import

To configure and run an import:

1. Configure the `config.xml` file:

   ```xml
   <import validateModel="validation" nonEmptyDatabaseAction="database_action" entityConflictAction="entity_action" />
   </import>
   ```

   where:
   - **validation**: Options are:
     - `true`: Validates the XML model before performing the import.
     - `false`: Does not validates the XML model before performing the import. (default)

   **Caution**: Oracle recommends you do not skip the model validation, so this parameter should always be set to `true`.

   - **database_action**: Options are:
     - `ignore`: The import completes even if it detects that the database is non-empty. (default)
     - `abort`: The import terminates if it detects that the database is non-empty.
     - `purge`: The import purges the existing database if it detects that the database is non-empty.

   If a model includes some changes for existing entities in the database, the import checks for order dependency. If those changes do not violate database constraints for pending orders, the import completes successfully. If the modified entities violate the existing orders, the violation is reported as known application exceptions with descriptive messages.

   - **entity_action**: The `entityConflictAction` import parameter value specifies the import behavior when the application encounters a conflict for an existing entity in the database. For example, a conflict occurs if you import a new
model that contains one or more entities that already exist in the database. Options are:

- **abort**: If an entity conflict exists, the import process stops.
- **replace**: The import replaces conflicted entities, that is, entities that already exist. (default)
- **ignore**: The import does not replace conflicted entities.

2. (Optional) Add the **selection** element within the **import** node to import targeted entities to an OSM cartridge using an XPath expression. Use the following syntax to define the selection element:

```xml
<import validateModel="validation" nonEmptyDatabaseAction="database_action" entityConflictAction="entity_action">
  <selection>/oms:model/oms:cartridge[@namespace="namespace" and @version="version"]/oms:entity</selection>
  <selection>/oms:model/oms:system_entity</selection>
</import>
```

where

- **namespace**: The namespace for the cartridge.
- **version**: The cartridge version.
- **entity**: The entity you are targeting. For example, workgroup, region, and schedule.

3. Do the following:

   a. If you using ant, run the following command:

   ```shell
   ant import
   ```

   b. If you using a batch script, run the following command:

   ```shell
   import.bat xmlModelFile config/config.xml
   ```

**Sample Procedure for Adding a New Workgroup Definition (Role)**

You can add a new workgroup definition (a workgroup is called a role in Design Studio) without having to redeploy a cartridge by using XMLIE.

---

**Note:** The workgroup definition is a system level entity that can be applied to multiple different cartridges. If you add a new workgroup using XMLIE, make sure you do not overwrite existing workgroup definitions.

---

To add a new workgroup definition using XMLIE:

1. In the `OSM_home\SDK\XMLImportExport\config\config.xml` file, add a selection element to an export node that targets the workgroupDefinition entity.

   For example:

   ```xml
   <export validateModel="false" exposeEntityID="false" layout="singleDocument" readOnlyFileAction="ignore">
     <selection>/oms:model/oms:workgroupDefinition</selection>
   </export>
   ```

2. Save and exit the file.
3. In the **OSM_home\SDK\XMLImportExport** folder, create an XML document to store the XML workgroup information from the cartridge you are targeting. For example: `workgroupDefinition.xml`.

4. Run the **OSM_home\SDK\XMLImportExport\export.bat** script:

   For example:
   ```
   export.bat C:\osminstall\SDK\XMLImportExport\workgroupDefinition.xml
   C:\osminstall\SDK\XMLImportExport\config\config.xml
   ```

5. When the export is complete, open the XML file containing the workgroup information and add a new `workgroupDefinition` element and all child elements. For example:

   ```xml
   xmlns:osm="http://xmlns.oracle.com/communications/ordermanagement/model"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   file://D:/OSMSTA~1/SDK/XMLIMP~1/models/OmsModel.xsd">
   <workgroupDefinition name="newWorkgroup">
   <description>newWorkgroup</description>
   <permissions>
   <createdVersionedOrders />
   <exceptionProcessing />
   <onlineReports />
   <priorityModification />
   <referenceNumberModification />
   <searchView />
   <taskAssignment />
   <worklistViewer />
   </permissions>
   <calendar>
   <weeklyWorkHours>no_schedule</weeklyWorkHours>
   <region>no_region</region>
   </calendar>
   </workgroupDefinition>
   </model>
   ```

6. Save and exit the file.

7. Run the **OSM_home\SDK\XMLImportExport\import.bat** script:

   For example:
   ```
   import.bat C:\osminstall\SDK\XMLImportExport\workgroupDefinition.xml
   C:\osminstall\SDK\XMLImportExport\config\config.xml
   ```
Sample Procedure for Adding a Task to a Workgroup (Role)

You can add a task to a workgroup (a workgroup is called a role in Design Studio) without having to redeploy a cartridge by using XMLIE.

To add a task to a workgroup using XMLIE:

1. In the OSM_home\SDK\XMLImportExport\config\config.xml file, add a selection element to an export node that targets the workgroup entity in the cartridge you want to add a task to.

For example:

```xml
<export validateModel="false" exposeEntityID="false" layout="singleDocument" readOnlyFileAction="ignore">
  <selection>/oms:model/oms:cartridge[@namespace="bb_ocm_demo" and @version="1.0.0.0.1"]/oms:workgroup</selection>
</export>
```

2. Save and exit the file.

3. In the OSM_home\SDK\XMLImportExport\ folder, create an XML document to store the XML workgroup information from the cartridge you are targeting. For example: worgroup.xml.

4. Run the OSM_home\SDK\XMLImportExport\export.bat script:

   For example:
   ```bash
   export.bat C:\osminstall\SDK\XMLImportExport\worgroup.xml
   C:\osminstall\SDK\XMLImportExport\config\config.xml
   ```

5. When the export is complete, open the XML file containing the workgroup information and add a new task element. For example:

   ```xml
   <?xml version = '1.0' encoding = 'ISO-8859-2'?>
   xmlns:osm="http://xmlns.oracle.com/communications/ordermanagement/model"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   file:///D:/OSMSTA~1/SDK/XMLIMP~1//models/OmsModel.xsd">
     <schemaVersion>7.2.0</schemaVersion>
   ```
<version>
  <label>7.2.0.0.366</label>
  <majorVersion>1.0</majorVersion>
</version>
<cartridge namespace="bb_ocm_demo" version="1.0.0.1">
  <description>BB OCM Demo</description>
  <default>true</default>
  <timestamp>2012-05-28T13:05:12</timestamp>
  <workgroup name="demo">
    <column name="Phone ">
      <path>/subscriber_info/primary_phone_number</path>
    </column>
    <column name="Name">
      <path>/subscriber_info/name</path>
    </column>
    <permissions>
      <orderEntry>
        <orderType>add_adsl_siebel</orderType>
        <orderSource>add_adsl_siebel</orderSource>
      </orderEntry>
      <task>activate_dslam</task>
      <task executionModes="do">add_adsl_siebel_creation</task>
      <task>add_capacity</task>
      <task>assign_port</task>
      <task>demo_query</task>
      <task>send_customer_survey</task>
      <task>ship_modem_self_install_pkg</task>
      <task>verify_adsl_service_availability</task>
      <task>verify_order</task>
      <task>new_task</task>
    </permissions>
  </workgroup>
</cartridge>

6. Save and exit the file.

7. Run the `OSM_home\SDK\XMLImportExport\import.bat` script:

   For example:

   import.bat C:\osminstall\SDK\XMLImportExport\workgroup.xml
   C:\osminstall\SDK\XMLImportExport\config\config.xml

---

**Note:** In this scenario, the XML model file contains only those elements that need to be imported. If the model file contained other elements that did not need to be imported, you can add a selection element to the import node in the `OSM_home\SDK\XMLImportExport\config\config.xml` file that targets the workgroup entity in the model file.

For example:

   <import validateModel="false"
         nonEmptyDatabaseAction="ignore"
         entityConflictAction="replace">
     <selection>/oms:model/oms:cartridge[namespace="bb_ocm_demo" and @version="1.0.0.1"]/oms:workgroup</selection>
   </import>
About Purging MetaData and Data

XMLIE provides the `purge.bat`, `purgeOrder.bat` scripts and ant `purge`, `undeploy`, and `forceundeploy` commands, which are used to purge metadata and data from an OSM schema. For example, you may need to purge an existing schema prior to importing a new model, or you may need to purge the data from a test environment at the beginning of each new phase of testing.

Using the `purge.bat` script or purge ant command, you can remove everything (all metadata and data) from the schema, or you can remove data from a specified cartridge. Using the `purgeOrder.bat` script or purge ant command, you can remove all the data. The `purgeOrder.bat` script and purge ant command do not affect metadata.

Oracle recommends that you purge all order data related to a cartridge before purging a cartridge. Purging large amounts of order data or cartridges should only be done during off-peak hours.

**Note:** You must shut down the WebLogic Server before executing the `purge` command or an exception is thrown.

The `purge.bat` script and purge ant command are not transactional so any unexpected failure may leave the schema in an invalid state. If this occurs, repeat the `purge.bat` script or purge ant command until it completes successfully.

Configuring and Running a Database Schema or Cartridge Purge

To purge the entire schema (metadata and orders) or undeploy a specific cartridge

1. If you are using ant commands, do one of the following:

   **Note:** You can target an entire schema for the `undeploy`, `forceundeploy`, and `purge` commands, or you can specify a namespace or a namespace and version by configuring the following `build.properties` parameters:

   - `xmlie.root.namespace`
   - `xmlie.root.version`

   For more information about these parameters, see "Configuring the `build.properties` File for Ant Commands".

   a. If you want to purge a target schema using ant, use the following command:
      
      ```
      ant purge
      ```

   b. If you want to undeploy a cartridge, but only if no pending orders exist using ant, use the following command:
      
      ```
      ant undeploy
      ```

   c. If you want to undeploy a cartridge, even if pending orders exist using ant, use the following command:
      
      ```
      ant forceUndeploy
      ```

2. If you are using batch scripts, do one of the following:
a. If you want to completely purge a target schema, use the following batch script and syntax:
   
   ```
   purge.bat config\config.xml force
   ```
   
   If you run this script without the `force` attribute, the `purge.bat` script fails if any pending orders exist on the cartridge you are purging. If you run this script with the `force` attribute, the script purges the cartridge and all pending orders.

b. If you want to undeploy every version of a cartridge, use the following batch script and syntax:
   
   ```
   purge.bat config\config.xml force -n namespace
   ```
   
   where `namespace` is the OSM cartridge namespace.

c. If you want to undeploy a specific cartridge version, use the following batch script and syntax:
   
   ```
   purge.bat config\config.xml force -n namespace -v version
   ```
   
   where `namespace` and `version` are the OSM namespace and cartridge version.

About Purging Orders

OSM provides the following ways to remove orders:

- XMLIE order purge - You can purge orders using criteria such as one or more order states, purge before date, order source, order type, namespace, version, start date, and stop date using XMLIE order purge.
- SQL script - You can remove in-flight orders from the system using SQL script.
- Dropping old partitions that contain completed orders.

**Note:** Oracle recommends dropping old partitions that contain completed orders as the best way to purge orders (see "Dropping Partitions with Oracle Scripter" for more information). If you cannot use this method because of pending orders, you can use the XMLIE order purge script as a slower alternative.

Deleting In-Flight Orders using SQL Scripts

An order can contain many tasks, with often hundreds running in parallel. This can make it difficult to cancel an order that is in progress. For example, you may need to cancel an in-flight order if incorrect data has been used.

A script is provided in the following directory: `OSM_home/Database/om/...`. Use this script to remove a running order from OSM (stop all tasks and complete the order without starting any subsequent tasks).

To delete in-flight orders, run the following SQL command:

```python
exec om_order_maintenance_pkg.remove_order(order_seq_id);
commit;
```

where `order_seq_id` is the sequence ID for the order you want to purge.
Purging Orders with the orderPurge.bat Script on Windows

You can configure the orderPurge.bat script to:

- purge orders based on one or more order states
- purge orders from a specific cartridge (namespace/version) or all cartridges
- schedule the purge to run during off-peak hours or run it immediately

Use the following commands to perform the indicated order purge operation.

1. Use the following syntax to run an immediate order purge:

   ```
   orderPurge.bat xmlConfigFile -p DBPassword doPurge "purge_before=before_purge" "order_states=order_states" "namespace=namespace" "version=version" "order_type=order_type" "order_source=order_source"
   ```

   where:

   - **before_purge**: Use this data parameter with the order_state parameter. For example, to purge orders completed 30 days ago, specify order_state="COMPLETED" and purge_before=2011-06-28T13:39:00 EST (or a date that is 30 days before the current date). Options are:
     - all: All orders that were created before this date are considered for the purge
     - any closed state: All orders whose completion date is before this date are considered for the purge.
     - any open state: All orders that were created and transitioned to the state specified before this date are considered for the purge.

   If no purge_before date is specified, the date is set to 5 seconds before the purge starts.

   - **order_states**: An order state must be specified. Options are one or more of the following comma separated values:
     - all
     - open
     - not_running
     - running
     - not_started
     - suspended
     - cancelled
     - compensating
     - amending
     - cancelling
     - closed

   Note: All date parameters must be specified in the format: `yyyy-mm-ddThh24:mi:ss`

   For example: `2011-06-28T13:39:00 EST`
About Purging MetaData and Data

- completed

For example: "not_started,completed"

- namespace: Must be specified. Valid values are either a namespace mnemonic or ALL (applies to all cartridges). For example, to purge all orders regardless of other conditions, specify `order_state="ALL"` and `namespace="ALL"`.

- version: If namespace is ALL, version is ignored. If namespace is specified but no version is specified, the purge applies to all versions of the namespace.

- order_type: The order type mnemonic. If specified, only orders with this type are purged.

- order_source: The order source mnemonic. If specified, only orders with this source are purged.

For example:

```bash
orderPurge.bat config/config.xml -p DBPassword doPurge "purge_before=2011-01-01T23:59:59 EST" "order_states=COMPLETED,NOT_STARTED" "namespace=abc" "version=1.0" "order_type=x" "order_source=y"
```

2. Use the following syntax to run a scheduled order purge:

```bash
orderPurge.bat xmlConfigFile -p DBPassword doPurge "purge_before=before_purge" "order_states=order_states" "namespace=namespace" "version=version" "order_type=order_type" "order_source=order_source" "stop_date=stop_date" "start_date=start_date"
```

where:

- `stop_date`: The time when the purge should stop, even if all orders satisfying the purge criteria have been purged (e.g., stop the purge before peak hours). If no `stop_date` is specified, the purge stops when all orders satisfying the purge criteria have been purged.

- `start_date`: For scheduled purges only - the time when the purge should start (must be later than the current time). When the `start_date` is reached, the purge starts automatically. If no `start_date` is specified, the purge is immediate.

For example:

```bash
orderPurge ./config/config.xml -p DBPassword schedulePurge "purge_before=2011-01-01T23:59:59 EST" "order_states=COMPLETED,NOT_STARTED" "namespace=abc" "version=1.0" "order_type=x" "order_source=y" "start_date=2007-01-01T23:59:59 EST"
```

3. Use the following syntax to list all scheduled order purges that have not started:

```bash
orderPurge.bat xmlConfigFile -p DBPassword listPurges
```

For example:

```bash
orderPurge.bat ./config/config.xml -p secret listPurges
```

4. Use the following syntax to remove an order purge that has not started:

```bash
orderPurge.bat xmlConfigFile -p DBPassword removePurge *job_id*
```
where \textit{job\_id} is the job ID of the scheduled purge.

For example:

\texttt{orderPurge.bat config/config.xml -p secret removePurge "job\_id=12345"}

\section*{Running Ant with the orderPurge.xml file On UNIX or Linux to Purge Orders}

To purge orders from an OSM schema using ant:

1. Open the \texttt{OSM\_home/SDK/XMLImportExport/build.properties} file.

2. If you want to perform an immediate purge of some or all orders before a certain date using the \texttt{immediateOrderPurge} ant \texttt{purge} command attribute, edit the following arguments:

\begin{verbatim}
xmlie.orderPurge.purgeBefore=before_purge
xmlie.orderPurge.orderStates=status
xmlie.orderPurge.namespace=namespace
xmlie.orderPurge.version=version
xmlie.orderPurge.orderType=order_type
xmlie.orderPurge.orderSource=order_source
\end{verbatim}

where

- \texttt{before\_purge}: Use this data parameter with the \texttt{order\_state} parameter. For example, to purge orders completed 30 days ago, specify \texttt{order\_state="COMPLETED"} and \texttt{purge\_before=2011-06-28T13:39:00 EST} (or a date that is 30 days before the current date). Options are:
  - \texttt{all}: All orders that were created before this date are considered for the purge.
  - \texttt{any closed state}: All orders whose completion date is before this date are considered for the purge.
  - \texttt{any open state}: All orders that were created and transitioned to the state specified before this date are considered for the purge.

If no \texttt{purge\_before} date is specified, the date is set to 5 seconds before the purge starts.

- \texttt{order\_states}: An order state must be specified. Options are one or more of the following comma separated values:
  - \texttt{all}
  - \texttt{open}
  - \texttt{not\_running}
  - \texttt{running}
  - \texttt{not\_started}
  - \texttt{suspended}
  - \texttt{cancelled}
  - \texttt{compensating}
  - \texttt{amending}
  - \texttt{cancelling}
About Purging MetaData and Data

- closed
- completed

For example: "not_started, completed"

- namespace: Must be specified. Valid values are either a namespace mnemonic or ALL (applies to all cartridges). For example, to purge all orders regardless of other conditions, specify `order_state="ALL"` and `namespace="ALL"`.

- version: If namespace is ALL, version is ignored. If namespace is specified but no version is specified, the purge applies to all versions of the namespace.

- order_type: The order type mnemonic. If specified, only orders with this type are purged.

- order_source: The order source mnemonic. If specified, only orders with this source are purged.

For example:

```
xmle.orderPurge.orderStates=COMPLETED
xmle.orderPurge.namespace=default
xmle.orderPurge.version=1.0.0.0.0
xmle.orderPurge.orderType=ot
xmle.orderPurge.orderSource=os
```

3. If you want to schedule a purge of some or all orders on a certain date using the `scheduleOrderPurge` ant purge command attribute, edit the following additional arguments:

```
xmle.orderPurge.startDate=start_date
xmle.orderPurge.stopDate=stop_date
```

where:

- `stop_date`: The time when the purge should stop, even if all orders satisfying the purge criteria have been purged (e.g., stop the purge before peak hours). If no stop_date is specified, the purge stops when all orders satisfying the purge criteria have been purged.

- `start_date`: For scheduled purges only - the time when the purge should start (must be later than the current time). When the start_date is reached, the purge starts automatically. If no start_date is specified, the purge is immediate.

For example:

```
xmle.orderPurge.orderStates=COMPLETED
xmle.orderPurge.namespace=default
xmle.orderPurge.version=1.0.0.0.0
xmle.orderPurge.orderType=ot
xmle.orderPurge.orderSource=os
xmle.orderPurge.startDate=2007-01-01T00:01:01 EST
xmle.orderPurge.stopDate=2007-12-31T23:59:59 EST
```

4. If you want to remove an scheduled order purge that has not started using the `removeOrderPurge` ant purge command attribute, edit the following arguments:

```
xmle.orderPurge.jobId=job_id
```
where *job_id* is the job ID of the scheduled purge.

For example:

```plaintext
xmlie.orderPurge.jobId=0
```

---

**Note:** Orders that satisfy the purge criteria are purged and related details such as the number of orders purged are logged in the XMLIE log file (as configured in config.xml). If an error occurs, the purge stops. Errors and exceptions are output to the command line and are logged in the log files.

---

5. Use the following syntax to run an immediate order purge:
   ```plaintext
   ant immediateOrderPurge
   ```

6. Use the following syntax to run a scheduled order purge:
   ```plaintext
   ant scheduleOrderPurge
   ```

7. Use the following syntax to list all scheduled order purges that have not started:
   ```plaintext
   ant listOrderPurges
   ```

8. Use the following syntax to remove an order purge that has not started:
   ```plaintext
   ant removeOrderPurge
   ```

---

### About Migrating Orders

XMLIE provides the **migrate** ant command and a **migrate.bat** script to migrate (copy) orders from one version of a cartridge to another. You can only migrate orders between versions of the same namespace, not between different namespaces. You can also migrate order data, reference numbers, remarks, attachments and specify a single order type/source to migrate, or all order types/sources within the cartridge.

For example, you may need to migrate orders within a cartridge when upgrading to a new version of OSM. Migrated orders are placed at the first process task, regardless of where the original order was in the process flow.

You cannot migrate orders across environments, for example, from a production environment to a test environment.

XMLIE migrates orders in three steps:

1. Create a new order and copy data from the source order to the target order.
   
   All order field data is copied provided the field has a value in the source order and the field is defined in the target order's creation task data. Order reference numbers and remarks/attachments are also copied if *copyReference* and *copyRemarks* are set to true. The migrated order is placed at the first process task, not at the equivalent task in the original order.

2. Once the copy is complete, close the source order (closeSource set to true).

   **Note:** If you choose to close the source order, the Exception Processing function must be associated with your workgroup.
3. Submit the target order (submitTarget set to true).

Orders are migrated individually. If an error occurs in any of these steps and errorAction is set to abort, processing stops immediately. Prior steps are not rolled back. If errorAction is set to ignore, any remaining steps are skipped and the process starts over at step 1 with the next available order to be migrated.

---

**Note:** Order migration should only be done within a window where no other order processing will occur.

---

It is extremely important that the target order creation task data contain all of the fields in the source order. The fields must be the same data type and have the same mnemonic and length to be considered equal. Any field that exists in the source order but not in the target creation task data is ignored. Fields that are defined in the target creation task data that have no associated data in the source order remain blank.

The most common failure scenarios are:

- The specified target version or type/source combination does not exist. This is a fatal exception because no orders can be migrated. Processing halts immediately regardless of the errorAction setting.
- The target order does not submit because the target creation task data contains mandatory fields that were not set in the source order.

### Configuring and Running an Order Migration

To configure and run an order migration:

1. Configure the config.xml file migration and userInteraction elements.

```xml
<config>
  <migration>
    <migrate submitTarget="submitTarget" closeSource="closeSource"
      copyReference="copyReference" copyRemarks="copyRemarks"
      errorAction="errorAction"/>
    <userInteraction confirmation="confirmation"/>
  </migration>
</config>
```

where:

- **submitTarget**: Options are:
  - **true**: Submits the target order following migration. (default)
  - **false**: Leaves the target order in the creation task.

- **closeSource**: Options are:
  - **true**: Closes the source order following migration. If you choose to close the source order, the Exception Processing function must be associated with your workgroup. (default)
  - **false**: Leaves the source order unchanged by the migration operation.

- **copyReference**: Options are:
  - **true**: Sets the order reference number of the target order to that of the source order. (default)
  - **false**: Leaves the target order reference number empty.

- **copyRemarks**: Options are:
  - **true**: Copies the source order remarks and attachments to the target order. (default)
About Migrating Orders

Using the XML Import/Export Application

1. false: Does not copy remarks and attachments.

■ errorAction: Options are:
  - abort: Stops processing immediately. (default)
  - ignore: Attempts to migrate the next available order.

■ confirmation: Used for validation while migrating, so if there are any warnings/errors it might ask user for confirmation. Options are:
  - true: Confirms the warning/error message if any.
  - false: Does not confirm the warning/error messages if any.

2. If you are using batch scripts, use the following syntax to migrate an order from one version of a cartridge to another version of the same cartridge:

   migrate.bat config/config.xml -sourcenamespace namespace -sourceversion version
   sourceordertype type -sourceordersource source -targetversion version

where:
  - namespace: Must be specified. Valid values are the namespace mnemonic.
  - version: Must be specified. The versions of the source namespace and the target namespace.
  - type: The order type mnemonic. If specified, only orders with this type are migrated.
  - source: The order source name. If specified, only orders with this source are migrated.

For example:

   migrate.bat config/config.xml -sourcenamespace default -sourceversion 1.0
   -targetversion 2.0

   and

   migrate.bat config/config.xml -sourcenamespace default -sourceversion 1.0
   -sourceordertype request for long distance -sourceordersource client care
   -targetversion 2.0

3. If you are using the ant migration command, do the following:

   a. Open the OSM_home/SDK/XMLImportExport/build.properties file.

   b. Edit the following arguments:

      xmlie.root.namespace=namespace
      xmlie.root.version=version
      xmlie.root.sourceordertype=type
      xmlie.root.sourceordersource=source
      xmlie.root.targetorderversion=version

where:
   - namespace: Must be specified. Valid values are the namespace mnemonic.
   - version: Must be specified. The versions of the source namespace and the target namespace.
   - type: The order type mnemonic. If specified, only orders with this type are migrated.
source: The order source name. If specified, only orders with this source are migrated.

For example:

xmlie.root.namespace=bb_ocm_demo
xmlie.root.version=1.0.0.0.0
xmlie.root.sourceordertype=Add Order
xmlie.root.sourceordersource=Add Order
xmlie.root.targetorderversion=1.0.0.0.1

c. Use the following syntax start a migration from one version of a cartridge to another version of a cartridge:

    ant migrate

About Validating the Metadata Model and Data

XMLIE provides the validate command, which is used to assure metadata model validity for an OSM schema. Using the validate command, you can perform the following:

- **Model validation**: The XML schema model defines most entity relationships and constraints.
- **Pending order validation**: This validation is applied during the import process (import.bat). It performs validations that have not been covered by validate.bat.

---

**Note**: When you perform a validation, you must supply a well-formed model, otherwise you may encounter undefined exceptions.

---

The application applies an XML schema validation as well as some additional rules that could not be implemented in the XML schema. You can use any XML application to validate the model against the schema, but for additional validations, the explicit validation (using validate.bat) or implicit validation (before import or after export), assures model correctness and completeness.

You can validate the metadata model before the import takes place, or change config.xml so that the validation takes place during the import.

Configuring and Running an XML Document Validation

To configure and run a validation:

1. Configure the config.xml file.

   `<validation validateAgainstDB="validateAgainstDB" validateDocument="validateDocument" validationReportURI="filename_path"/>

where:

- **validateAgainstDB**: Options are:
  - **true**: Validates the XML document against existing orders in the database schema to ensure it is compatible. (default)
  - **false**: Does not perform an XML model validation against the database schema.
About Creating a Graphical Representation of the Metadata Model

- **validateDocument**: Options are:
  - **true**: Validates the XML document against the OSM XML schema to ensure it is well formed. (default)
  - **false**: Does not perform the XML document validation.

- **filename_path** is the path to the validation log file that includes the file name of the validation log file.

2. Do one of the following:
   a. If you are using ant commands, do the following:
      ```
      ant validate
      ```
   b. If you are using batch scripts, do the following:
      ```
      validate.bat xmlModelFile config\config.xml
      ```

**Validating an XML Document During the Import or Export Process**

To validate an XML document *during* the import or export process, set the configuration file **validateModel** parameter to **true**. See "Configuring and Running an Import" and "Configuring and Running an Export" for syntax information and examples.

**About Creating a Graphical Representation of the Metadata Model**

XMLIE provides the **modeldoc** command, which is used to convert an OSM metadata XML document into an HTML presentation. The HTML presentation is a graphical representation of the metadata model which is much easier to understand and navigate than the metadata XML document. You can use the resulting HTML presentation to view the relationships between and dependencies of the various metadata elements using standard HTML presentation navigation methods.

If you are going to use the **modeldoc.bat** script or the **ant htmlModel** command to create HTML representations of cartridge metadata, some configuration needs to be set up in order for GraphViz to support these commands and scripts.

**Configuring and Creating a Graphical Representation of a Metadata Model**

To configure and create a graphical representation of a metadata model:

1. Download and install GraphViz. For more information see *OSM Installation Guide*.
2. Update the **OSM_home\SDK\XMLImportExport\config.bat** script with an environment variable containing the path to the third-party GraphViz library. For example:
   ```
   set GRAPHVIZ=C:\Applications\ATT\Graphviz\bin
   ```
3. If you are using the **OSM_home\SDK\XMLImportExport\modeldoc.bat** script, update the Java command line in the **modeldoc.bat** file with an entry referring to the environment variable containing the path to the third-party GraphViz library:
   ```
   %JAVA_HOME%/bin/java %JAVA_OPTS% -classpath %CLASSPATH% %APP_PROPERTIES% com.mslv.cms.metadatahandler.operation.ModelDocOperator %XML_MODEL% %GRAPHVIZ% ./modeldoc
4. If you are using the ant `htmlModel` command, configure the `graphiz` property in the `OSM_home\SDK\XMLImportExport\build.xml` file to specify the directory where the third-party GraphViz software is installed. For example:

   `<property name="graphiz" value="/bin/ATT/Graphviz/bin"/>

5. If you are using batch scripts, run the following script:

   `modeldoc.bat xmlModelFile HTMLModelDirectoryPath`

   where `HTMLModelDirectoryPath`: Specifies the path to the directory for the HTML model files for the `modeldoc.bat` script.

6. If you are using the ant `htmlModel` command, run the following command:

   `ant htmlModel`

**Viewing the Graphical Representation**

To view the HTML presentation, open the resulting HTML index file and begin navigating through the metadata model using the automatically generated hyperlinks.

---

**Note:** To view the HTML presentation, your browser must support Adobe SVG Viewer.

---
This chapter describes how to configure time zone settings in Oracle Communications Order and Service Management (OSM). This is an optional configuration task.

**Configuring Time Zone Settings**

The `oms_timezone` parameter, in the `om_parameter` table of the database, must be set to the non-dst timezone. This setting must be defined in seconds offset to GMT for the time zone where your OSM database resides. For example, if you are setting this parameter for Eastern Standard Time (EST), use the value -18000 (60 seconds x 60 minutes = 3600 seconds—or one hour—x 5 for five hours offset from GMT). Time zones east of GMT use positive numbers, and time zones west of GMT use negative numbers.

This setting must be done to the OSM database/schema at creation time. The parameter is called "oms_server_tz_offset_seconds" in the Scripter. This value is used by both the OSM database and OSM Administrator.

The `database_timezone_offset`, in the `oms-config.xml` file of the OSM server, must be set to be exactly the same as the value set in 1. This value is used by the OSM server.
This chapter provides guidelines to help you troubleshoot problems with your Oracle Communications Order and Service Management (OSM) system.

**General Checklist for Resolving Problems with OSM**

If you have a problem with your OSM system, go through the following checklist before you contact Oracle Technical Support:

- What exactly is the problem? Can you isolate it? For example, if an order causes a problem on one computer, does it give the same result on another computer?
  
  Oracle Technical Support needs a clear and concise description of the problem, including when it began to occur.

- What do the log files say?
  
  This is the first thing that Oracle Technical Support asks for. Check the error log for the OSM component you are having problems with.

- Have you read the documentation?
  
  Look through the list of common problems and their solutions in "Diagnosing Some Common Problems with OSM".

- Has anything changed in the system? Did you install any new hardware or new software? Did the network change in any way? Does the problem resemble another one you had previously? Has your system usage recently jumped significantly?

- Is the system otherwise operating normally? Has response time or the level of system resources changed? Are users complaining about additional or different problems?

**Diagnosing Some Common Problems with OSM**

This section describes common problems and their solutions.

**Cannot Log In or Access Certain Functionality**

If you cannot log in or access certain functionality, check the following possible causes:

- Are you a valid user in the Oracle WebLogic Server security realm?

- Is the OSM Web application deployed?

- Are all OSM Enterprise Java Beans (EJB) deployed?
Are the OSM database resources deployed?
Do you belong to the correct groups in the WebLogic Server security realm?
Do you belong to any OSM workgroup?

System Appears Slow

If the functionality of OSM appears to be present, but performance is slow, check the following possible causes:

- The amount of memory being used (check the max memory configuration in the WebLogic server startup script on the workstation where you have deployed OSM)
- The CPU and disk usage on the machine hosting the OSM database
- The database connections
- For slow worklist access, check the number of flexible headers on your worklist. The number of flexible headers has a direct negative effect on worklist performance.

To Avoid Timeout Issues in Design Studio

To avoid timeout issues from Oracle Communications Design Studio during cartridge deployment, set the following parameter in Domain_HOME/bin/startWebLogic.sh after the line that reads umask 037:

```
export JAVA_OPTIONS="-Dcom.mslv.oms.cartridgemgmt.DeployCartridgeMDB.CartridgeDeploymentTransactionTimeout=2400"
```

Also, you should increase the JTA value in the WebLogic Administration Console:

1. Log in to the WebLogic Administration Console.
2. In the Domain Configurations section, click Domain.
   The settings for your domain are displayed on a tabbed page.
3. Click the JTA tab and set the value of the Timeout Seconds parameter to 9000.
4. Click the Security tab and select Anonymous Admin Lookup Enabled.
5. Click Save.

**Note:** If an order has stopped processing due to a timeout, the order must be aborted. It will not resume processing even after the timeout value has been increased.

Error: "Login failed. Please try again."

If the error "Login failed. Please try again" is displayed when trying to log in through the Web client and you have entered the correct user name and password, you probably do not belong to the correct groups in the WebLogic Server security realm.

**Solution**

Log in to the WebLogic Administration Console using the administrator account. Make sure you have been added to the group OMS_client. Try to log in again.
No Users or Groups Are Displayed

After OSM installation, you do not see any users or groups on the Users and Groups tab. This is because non-dynamic changes have been made, and the Admin Server (and managed server, if applicable) requires a restart.

Solution
1. Restart the Admin/Managed Server to clear the condition. If the condition does not clear, proceed with the steps below.
2. Log in to the WebLogic Administration Console and select Domain.
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 appear.

Automation Plug-ins Are Not Getting Called

If the custom automation plug-ins are not getting called, check the following possible causes:
- Is the Automation configuration deployed properly?
- Are the JMS resources deployed?
- Are the JMS destinations, queues, and topics configured properly?

Too Many Open Files

If you have a large number of external clients connected to OSM and receive the error: "java.net.SocketException: Too many open files", do one of the following:
- From the WebLogic Administration Console, select Servers, then Server, then Protocols, and then HTTP. Reduce the value in Duration from the default 30 seconds to 15 or even 5 seconds. This will allow the WebLogic server to close idle HTTP connections and release more sockets.

Proxy Fails on a Clustered System

If a proxy fails on a clustered OSM system, all HTTP requests that would normally go through the proxy can no longer get to the OSM server. The problem could be with the physical host the server is running on, or it could be a problem with a standalone managing server that is not part of the cluster but is part of the domain.

To recover, restart the proxy.

Problems Displaying Gantt Charts on Solaris 5.10 Hosted Systems

When running on Solaris 5.10 only, to use X server to display Gantt charts in the Task Web client, you must configure the Java settings for the Oracle WebLogic Server to avoid display problems and system instability and performance problems.
See the discussion about enabling graphical displays in the post-installation section of the *OSM Installation Guide*.

**OSM Fails to Process Orders Because of Metadata Errors**

Metadata errors can occur in any cartridge with orchestration model entities and can cause order processing failures. Search for the string `Metadata Errors` in the Console view of the Cartridge Management editor in Design Studio. If you are not using Design Studio to deploy cartridges, look in the WebLogic Server logs for the same string.

For more information, see the discussion of metadata errors in *OSM Developer’s Guide*.

**Error: "No Backend Servers Available"**

If the error "No Backend Servers Available" is displayed, you are likely disconnected from your servers. Ensure your servers are connected and functional before continuing with OSM operations.

**Quick Fix Button Active During Order Template Conflicts in Design Studio**

Conflicts can occur when order templates are created in Design Studio. Presently, Quick Fix does not work for order template conflicts, even if the Quick Fix button is active. All order template conflicts must be resolved manually.

**Getting Help with OSM Problems**

If you cannot resolve your problems with OSM, contact Oracle Technical Support.

**Before You Contact Support**

Problems can often be fixed by shutting down OSM and restarting the computer that OSM runs on.

If that does not solve the problem, the first troubleshooting step is to look at the error log for the application or process that reported the problem. Consult *"General Checklist for Resolving Problems with OSM"* before reporting the problem to Oracle.

**Reporting Problems**

If *"General Checklist for Resolving Problems with OSM"* does not help you to resolve the problem, write down the pertinent information:

- A clear and concise description of the problem, including when it began to occur.
- Relevant portions of the log files.
- Relevant configuration files, such as `oms-config.xml`.
- Recent changes in your system, even if you do not think they are relevant.
- List of all the OSM components and patches installed on your system.

When you are ready, report the problem to Oracle.
This chapter details the Oracle Communications Order and Service Management (OSM) log messages. The sections included in this chapter are:

- OSM Catalog Messages
- Automation Catalog Messages

**OSM Catalog Messages**

*Table 14–1* shows OSM Catalog messages.
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600000     | error    | **Message Body**  
Order type/source is not found. The order type={1} /source={0} either does not exist, or is not available to the user.  
**Message Details**  
The order type={1}/ source={0} either does not exist, or is not available to the user.  
**Method**  
logSourceTypeNotFound(String source, String type) |
| 600001     | error    | **Message Body**  
Order is not found. The order with orderid={0} and orderHistId={1} does not exist, or is not available to the user.  
**Message Details**  
The order with orderid={0} and orderHistId={1} does not exist, or is not available to the user.  
**Cause**  
The orderHistId might not be up to date.  
**Action**  
Refresh server.  
**Method**  
logOrderNotFound(String orderId, String orderHistId) |
| 600002     | error    | **Message Body**  
Order template is not found. The order template does not exist, or is not available to the user.  
**Message Details**  
The order template does not exist, or is not available to the user.  
**Method**  
logOrderTemplateNotFound() |
| 600003     | error    | **Message Body**  
Remark is not found. The given remark (orderid={0}, histid={1} remarkid={2}) does not match a remark in OMS.  
**Message Details**  
The given remark (orderid={0}, histid={1} remarkid={2}) does not match a remark in OMS.  
**Cause**  
Remark might have been deleted from the server or cannot be found in a specified location.  
**Action**  
Contact your local administrator.  
**Method**  
logRemarkNotFound(String orderId, String orderHistId, String remarkId) |
### OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600004     | error    | **Message Body**  
Header for mnemonic path is not found. The header for mnemonic path={0} does not exist, or is not available to the user.  
**Message Details**  
The header for mnemonic path={0} does not exist, or is not available to the user.  
**Method**  
logHeaderNotFound(String mnemonicPath) |
| 600005     | error    | **Message Body**  
The format of the order data is not correct. The message details the error location.  
Invalid data = {0}  
**Message Details**  
Invalid data = {0}  
**Method**  
logOrderDataInvalid(String orderdata) |
| 600006     | error    | **Message Body**  
An attempt to update an order was made without first retrieving the order with an Accept parameter of true. Order (orderid={0} histid={1}) has not been accepted by user={2}.  
**Message Details**  
Order (orderid={0} histid={1}) has not been accepted by user={2}  
**Cause**  
An attempt to update an order was made without accepting the order.  
**Action**  
You should accept the order first, then update the order.  
**Method**  
logOrderNotAcceptedByUser(String orderId, String orderHistId, String userid) |
| 600007     | error    | **Message Body**  
Order update failed. The order (orderid={0} histid={1}) could not be updated due to a data format error. The message details the reason for failure. Data={2}  
**Message Details**  
The order (orderid={0} histid={1}) could not be updated due to a data format error. The message details the reason for failure. Data={2}  
**Causes**  
Data format error.  
**Action**  
Make sure all your data are in correct format and comply with their masks.  
**Method**  
logOrderUpdateFailed(String orderId, String histid, String data) |
Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600008     | error    | Message Body
Mandatory check failed. A mandatory field was not given a value when attempting to create, assign, complete, or suspend an order. {2} number of data was missing for order with order id={0}, order history id={1}. The first missing/extra node is node id={3} and order type={4}.

Message Details
Mandatory check failed. A mandatory field was not given a value when attempting to create, assign, complete, or suspend an order. {2} number of data was missing for order with order id={0}, order history id={1}. The first missing/extra node is node id={3} and order type={4}

Cause
Not all mandatory fields are filled.

Action
Fill in data for all mandatory fields.

Method
logMandatoryCheckFailed(String orderID, String orderHistID, String num, String firstNodeID, String firstNodeType)

| 600009     | error    | Message Body
Transition is invalid. The order (orderid={0} histid={1}) cannot be transitioned to state={2}. Use ListStatesNStatuses.Request to get a list of valid states.

Message Details
The order (orderid={0} histid={1}) cannot be transitioned to state={2}. Use ListStatesNStatuses.Request to get a list of valid states.

Cause
Cannot transition to the selected state.

Action
Use the ListStatesNStatuses XML API request to get a list of valid states. See OSM Developer’s Guide for information.

Method
logTransitionInvalid(String orderid, String histid, String state)

| 600010     | error    | Message Body
Unable to accept order. When retrieving an order for update, the order (orderid={0} histid={1}) cannot be accepted by the current user={2}

Message Details
When retrieving an order for update, the order (orderid={0} histid={1}) cannot be accepted by the current user={2}.

Cause
The orderid and histid are not up to date, or the order has been accepted by another user.

Action
Refresh the server to get new orderid and histid. If the order is currently accepted by another user, you cannot perform Accept operation.

Method
logUnableToAccept(long orderid, long histid, String userid)
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600011     | error    | **Message Body**  
User is not found. The order orderid={0} histid={1} cannot be assigned to userid={2}.  
**Message Details**  
The order orderid={0} histid={1} cannot be assigned to userid={2}.  
**Cause**  
User is not found.  
**Action**  
Try to assign the order to another user.  
**Method**  
logUserNotFound(String orderid, String histid, String userid) |
| 600012     | error    | **Message Body**  
Invalid state mnemonic. The order (orderid={0} histid={1}) cannot be suspended with the given state (state={2}) mnemonic. Note: Only user-defined states are valid. If you want to complete or assign an order, you must use the appropriate request.  
**Message Details**  
The order (orderid={0} histid={1}) cannot be suspended with the given state (state={2}) mnemonic. Note: Only user-defined states are valid. If you want to complete or assign an order, you must use the appropriate request.  
**Cause**  
You are calling suspendOrder with an invalid user defined state.  
**Action**  
Supply a valid user defined state, or try to use other appropriate requests.  
**Method**  
logInvalidStateMnemonic(String orderid, String histid, String state) |
| 600013     | error    | **Message Body**  
Invalid status mnemonic. The order (orderid={0} histid={1}) cannot be completed with the given status mnemonic (status={2}).  
**Message Details**  
The order (orderid={0} histid={1}) cannot be completed with the given status mnemonic (status={2}).  
**Cause**  
The status might not be valid for the current task.  
**Action**  
Supply a valid status.  
**Method**  
logInvalidStatusMnemonic(String orderid, String histid, String status) |
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
<th>Message Body</th>
</tr>
</thead>
<tbody>
<tr>
<td>600014</td>
<td>error</td>
<td></td>
<td>Remark cannot be modified. The time interval in which a created remark can be modified has elapsed. The remark can no longer be modified. (orderid={0} histid={1} remarkid={2} userid={3})</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message Details</td>
<td>Remark cannot be modified. The time interval in which a created remark can be modified has elapsed. The remark can no longer be modified. (orderid={0} histid={1} remarkid={2} userid={3})</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Method</td>
<td>logRemarkCannotBeModified(String orderid, String histid, String remarkid, String userid)</td>
</tr>
<tr>
<td>600015</td>
<td>error</td>
<td></td>
<td>Request Unknown. The request type could not be identified. Type given={0}</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message Details</td>
<td>The request type could not be identified. Type given={0}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Method</td>
<td>logRequestUnknown(String type)</td>
</tr>
<tr>
<td>600016</td>
<td>error</td>
<td></td>
<td>Request parameter error. A parameter for the request is missing or invalid. The message details the parameter in question Parameter = {0}, request type = {1}.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message Details</td>
<td>A parameter for the request is missing or invalid. The message details the parameter in question Parameter = {0}, request type = {1}.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Method</td>
<td>logRequestParameterError(String parameter, String requestType)</td>
</tr>
<tr>
<td>600017</td>
<td>error</td>
<td></td>
<td>Not authorized. The user={0} is not authorized to make the request={1}.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Message Details</td>
<td>The user={0} is not authorized to make the request={1}.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cause</td>
<td>The user is not authorized to perform the operation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Method</td>
<td>logNotAuthorized(String userid, String request)</td>
</tr>
<tr>
<td>600018</td>
<td>error</td>
<td></td>
<td>Database connection failed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Method</td>
<td>logDatabaseConnectionFailed()</td>
</tr>
<tr>
<td>600019</td>
<td>error</td>
<td></td>
<td>Security violation by user = {0}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cause</td>
<td>User is not authorized to perform this operation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Method</td>
<td>logSecurityViolation(String userid)</td>
</tr>
</tbody>
</table>
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>600020 error</td>
<td>Message Body</td>
<td>Naming exception was thrown while looking up JNDI name={0}.</td>
</tr>
<tr>
<td></td>
<td>Cause</td>
<td>JNDI name might not exist.</td>
</tr>
<tr>
<td></td>
<td>Action</td>
<td>Contact your local administrator.</td>
</tr>
<tr>
<td></td>
<td>Method</td>
<td>logNamingException(String name)</td>
</tr>
<tr>
<td>600021 error</td>
<td>Message Body</td>
<td>Remote exception thrown, while access object = {0}</td>
</tr>
<tr>
<td></td>
<td>Message Details</td>
<td>Remote exception was thrown while working with an EJB.</td>
</tr>
<tr>
<td></td>
<td>Method</td>
<td>logRemoteException(String name)</td>
</tr>
<tr>
<td>600022 error</td>
<td>Message Body</td>
<td>EJB Create exception thrown while creating object={0}.</td>
</tr>
<tr>
<td></td>
<td>Message Details</td>
<td>EJB Create exception thrown.</td>
</tr>
<tr>
<td></td>
<td>Method</td>
<td>logEJBCreateException(String name)</td>
</tr>
<tr>
<td>600023 error</td>
<td>Message Body</td>
<td>Unknown exception thrown, message={0}.</td>
</tr>
<tr>
<td></td>
<td>Message Details</td>
<td>Unknown exception thrown, message={0}.</td>
</tr>
<tr>
<td></td>
<td>Method</td>
<td>logUnknownException(String message)</td>
</tr>
<tr>
<td>600024 error</td>
<td>Message Body</td>
<td>Cannot deliver JMS message to queue = {0}</td>
</tr>
<tr>
<td></td>
<td>Message Details</td>
<td>Cannot deliver JMS message.</td>
</tr>
<tr>
<td></td>
<td>Cause</td>
<td>JMS Queue might be down.</td>
</tr>
<tr>
<td></td>
<td>Method</td>
<td>logCannotDeliverMessage(String arg)</td>
</tr>
<tr>
<td>Error Code</td>
<td>Severity</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 600025     | error    | Message Body  
Invalid XML document, doc={0}.  
Message Details  
Invalid XML document, doc={0}.  
Cause  
XML document has syntax errors.  
Action  
Fix the syntax errors.  
Method  
logInvalidXMLDocument(String doc) |
| 600028     | debug    | Message Body  
SQL: {0}  
Message Details  
SQL statement execution  
Method  
logSQL(String sql) |
| 600029     | debug    | Message Body  
[0]  
Message Details  
Log of a request with all its parameters, based on a toString()  
Method  
logRequestWithParameters(com.nortel.oms.request.Request request) |
| 600030     | debug    | Message Body  
Creating order node: mnemonic path {0}, node id {1}, node type {2}, node index {3}, parent index {4}, double value {5}, value {6}, parent new? {7}  
Message Details  
Node information was sent to the database  
Cause  
An order node has been created  
Action  
None  
Method  
logNodeCreate(String mnemonicPath, long nodeID, String nodeType, long nodeIndex, long parentIndex, double doubleValue, String value, String parentNew) |
### OSM Catalog Messages

#### Table 14–1  (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600031     | debug    | **Message Body**
|            |          | Deleting Order Node: node ID {0}, node type {1}, node index {2}, old double value {3}, old text value {4}  |
|            |          | **Message Details**
|            |          | Node information was sent to the database |
|            |          | **Cause**
|            |          | An order node has been deleted |
|            |          | **Action**
|            |          | None |
|            |          | **Method**
|            |          | logNodeDelete(long nodeID, String nodeTypeCode, long nodeIndex, double oldDoubleValue, String oldValue) |
| 600032     |          | **Message Body**
|            |          | Updating Order Node: mnemonic path {0}, node ID {1}, node type {2}, node index {3}, old double value {4}, old text value {5}, new double value {6}, new text value {7}  |
|            |          | **Message Details**
|            |          | Node information was sent to the database |
|            |          | **Action**
|            |          | None |
|            |          | **Method**
|            |          | logNodeUpdate(String mnemonicPath, long nodeID, String nodeTypeCode, long nodeIndex, double oldDouble, String oldText, double newDouble, String newText) |
| 600035     | debug    | **Message Body**
|            |          | Data validation failed in order editor. Node (name={0} nodeid={1} nodetype={2} nodeIndex={3} nodeDataType={4} mask={5}) does not comply with mask. Order ID={6} and user={7}  |
|            |          | **Message Details**
|            |          | Node (name={0} nodeid={1} nodetype={2} nodeIndex={3} nodeDataType={4} mask={5}) does not comply with mask. Order ID={6} and user={7} |
|            |          | **Cause**
|            |          | Data supplied do not comply with their masks. |
|            |          | **Action**
|            |          | Must supply data with correct format. |
|            |          | **Method**
|            |          | logOrderEditorDataValidationFailed(String name, String nodeId, String nodeType, String nodeIndex, String nodeDataType, String mask, String orderID, String user) |
| 600036     | debug    | **Message Body**
|            |          | Order editor - Create a node. |
|            |          | **Message Details**
|            |          | Order editor: create a node (nodeId={0} nodeType={1} parentWebID={2}) |
|            |          | **Method**
|            |          | logOrderEditorCreateNode(String nodeId, String nodeType, String parentWebID) |
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600037     | debug    | Message Body
Order editor - delete a node
Message Details
Order editor - delete a node (webID={0})
Method
logOrderEditorDeleteNode(String webID) |
| 600038     | debug    | Message Body
Notification Engine - create message.
Message Details
Creation of event message
Method
logNotificationEngineCreateMsg(String arg) |
| 600039     | debug    | Message Body
Could remove an event from the DB. Event Id = {0}
Method
logEventEngineRemoveEvent(long eventID) |
| 600055     | warning  | Message Body
An exception occurred while removing the session for user {0}. Reason: {1}.
Message Details
While logging out user {0}, an exception was thrown when calling EJBObject.remove().
Cause
The server does not allow the session to be removed, or a communication error occurred.
Method
logEJBRemoveException(String username, Throwable th) |
| 600063     | error    | Message Body
Error loading screen definitions from {0}. Reason: {1}.
Message Details
The screen definition file {0} could not be loaded. The screen definition file is required to construct the Web pages.
Cause
The OMS application has not been deployed or built properly.
Action
Make sure that the screendefinitions.xml file is in the oms.ear file:
oms.ear/oms.war/WEB-INF/conf/
Method
logCouldNotLoadScreenDefinitions(String url, Throwable th) |
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>600066</td>
<td>warning</td>
<td><strong>Message Body</strong>&lt;br&gt;Poller cannot locate listener {0}. Reason: {1}.&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;Cause&lt;br&gt;Poller Servlet was deployed before listener {0}.&lt;br&gt;Action&lt;br&gt;None&lt;br&gt;Method&lt;br&gt;logGetPollerListener(String listener, Throwable th)</td>
</tr>
<tr>
<td>600069</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;Could not unsubscribe for {0} event.&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;Event unsubscription for {0} event has failed.&lt;br&gt;Cause&lt;br&gt;JMS queue might be down.&lt;br&gt;Action&lt;br&gt;Use the WebLogic Server Console to verify JMS deployment.&lt;br&gt;Method&lt;br&gt;logEventTypeUnsubscribeException(String eventType)</td>
</tr>
<tr>
<td>600071</td>
<td>debug</td>
<td><strong>Message Body</strong>&lt;br&gt;Filter value {1} does not have the proper format ({3}) for {0}.&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;A filter could not be generated using the value {0}, formatter for {1} and mask {3}&lt;br&gt;Cause&lt;br&gt;User error&lt;br&gt;Action&lt;br&gt;Expected. You must retry.&lt;br&gt;Method&lt;br&gt;logFilterFormatError(String headerName, String filterValue, String mask)</td>
</tr>
<tr>
<td>600072</td>
<td>debug</td>
<td><strong>Message Body</strong>&lt;br&gt;{3} invalid. Wildcards are not permitted for operation {1} for header {0}&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;{3} invalid. Wildcards are not permitted for operation {1} for header {0}&lt;br&gt;Cause&lt;br&gt;User input. Expected error.&lt;br&gt;Action&lt;br&gt;You must retry.&lt;br&gt;Method&lt;br&gt;logFilterWildcardError(String headerName, String operationType, String value)</td>
</tr>
</tbody>
</table>
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600073     | error    | **Message Body**  
SSL is not enabled.  
**Message Details**  
SSL is not enabled.  
**Cause**  
SSL is not enabled.  
**Action**  
You must have SSL enabled for your server through WebLogic console.  
**Method**  
logSSLDisabled() |
| 600076     | error    | **Message Body**  
Unable to connect to remote file system for accessing attachments.  
**Message Details**  
Unable to connect to T3 remote file system for accessing attachments.  
**Cause**  
Targeted server might be down.  
**Action**  
Contact your local administrator.  
**Method**  
logT3ConnectionException() |
| 600077     | error    | **Message Body**  
Unable to add attachment with id={0} in remote file system.  
**Message Details**  
Unable to add attachment with id={0} in remote file system using T3 file services.  
**Cause**  
IO Exception.  
**Method**  
logT3AddAttachmentException(String attachmentID) |
| 600078     | error    | **Message Body**  
Unable to delete attachment with id={0} in remote file system. Attachment with id={0} does not exist.  
**Message Details**  
Unable to delete attachment with id={0} in remote file system using T3 file services. Attachment with id={0} does not exist.  
**Cause**  
The attachment does not exist.  
**Method**  
logT3DeleteAttachmentNotFoundException(String attachmentID) |
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600079     | error    | **Message Body**  
Unable to read attachment with id={0} in remote file system.  
**Message Details**  
Unable to read attachment with id={0} in remote file system using T3 file services.  
**Method**  
logT3ReadAttachmentException(String attachmentID) |
| 600086     |          | **Message Body**  
Retrieve PendingOrdersReport  
**Message Details**  
Retrieve PendingOrdersReport with orderTypesID={0}, orderSourceID={1}, summaryLevel={2}  
**Method**  
logPendingOrdersReport(String orderTypesID, String orderSourceID, String summaryLevel) |
| 600088     | error    | **Message Body**  
Unable to add attachment with id={0} in remote file system. Attachment with id={0} already exists.  
**Message Details**  
Unable to add attachment with id={0} in remote file system using T3 file services. Attachment with id={0} already exists.  
**Cause**  
The attachment already exists.  
**Action**  
None  
**Method**  
logT3AddAttachmentAlreadyExists(String attachmentID) |
| 600089     | error    | **Message Body**  
Unable to add attachment with id={0} in remote file system. Attachment with id={0} exceeds maximum file size specified in the configuration file.  
**Message Details**  
Unable to add attachment with id={0} in remote file system using T3 file services. Attachment with id={0} exceeds maximum file size specified in the configuration file.  
**Cause**  
Your attachment size is too big.  
**Action**  
Contact your local administrator.  
**Method**  
logT3AddAttachmentExceedMaxFileSize(String attachmentID) |
### Table 14–1  (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>600090</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;Unable to read attachment with id={0} in remote file system. Attachment with id={0} does not exist.&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;Unable to read attachment with id={0} in remote file system using T3 file services. Attachment with id={0} does not exist.&lt;br&gt;<strong>Cause</strong>&lt;br&gt;The attachment does not exist.&lt;br&gt;<strong>Action</strong>&lt;br&gt;None&lt;br&gt;<strong>Method</strong>&lt;br&gt;logT3ReadAttachmentNotFoundException(String attachmentID)</td>
</tr>
<tr>
<td>600091</td>
<td></td>
<td><strong>Message Body</strong>&lt;br&gt;SSL port is not found.&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;SSL port is not found.&lt;br&gt;<strong>Cause</strong>&lt;br&gt;Either server is not found, or there is no SSL port set up for the current server.&lt;br&gt;<strong>Action</strong>&lt;br&gt;Make sure SSL port was set up through WebLogic console.&lt;br&gt;<strong>Method</strong>&lt;br&gt;logSSLPortNotFoundException()</td>
</tr>
<tr>
<td>600092</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;MBean home is not found.&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;MBeanHome.ADMIN_JNDI_NAME cannot be found.&lt;br&gt;<strong>Cause</strong>&lt;br&gt;MBeanHome.ADMIN_JNDI_NAME cannot be found.&lt;br&gt;<strong>Action</strong>&lt;br&gt;None&lt;br&gt;<strong>Method</strong>&lt;br&gt;logMBeanHomeDisabled()</td>
</tr>
</tbody>
</table>
Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600093     | error    | **Message Body**  
DataSource connection to database could not be found.  
**Message Details**  
The EJB requesting a connection to the database does not have a DataSource configured. This caused the data access object to fail.  
**Cause**  
The database could not be located on server startup, or the EJB deployment descriptor ejb-jar.xml is missing a resource-ref to jdbc/DataSource, or WebLogic-ejb-jar.xml is missing a resource-descriptor tag to ordermanager/oms1/internal/jdbc/DataSource  
**Action**  
Determine if the database is available. Determine if EJB has correct deployment descriptor  
**Method**  
logDataSourceNotFound() |
| 600098     | error    | **Message Body**  
The current server name cannot be found  
**Message Details**  
The current server name cannot be found  
**Cause**  
MBean cannot find current run time server  
**Action**  
None  
**Method**  
logserverNameNotFound() |
| 600099     | error    | **Message Body**  
Get max_read_only_field_length property from oms-config.xml has failed.  
**Message Details**  
Get max_read_only_field_length property from oms-config.xml has failed.  
**Cause**  
max_read_only_field_length entry does not exist in oms-config.xml.  
**Action**  
Make sure max_read_only_field_length entry exists in oms-config.xml.  
**Method**  
logGetMaxReadOnlyLengthError() |
| 600103     | error    | **Message Body**  
OMS is not enabled to send {0} event to the automated agent.  
**Cause**  
Either it is disabled in oms-config.xml file or the {1} key is not found in the file.  
**Action**  
Make sure it is {1} key in the oms-config.xml file is set to true.  
**Method**  
logDisabledJMSEvent(String event, String key) |
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>600104</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;Exception thrown while trying to enable or disable (0) event.&lt;br&gt;<strong>Method</strong>&lt;br&gt;logExceptionFindListenerForJmsevent(String event)</td>
</tr>
<tr>
<td>600105</td>
<td>warning</td>
<td><strong>Message Body</strong>&lt;br&gt;Cannot send messages. JMS connection is down.&lt;br&gt;<strong>Cause</strong>&lt;br&gt;JMS connection is down.&lt;br&gt;<strong>Action</strong>&lt;br&gt;Use the WebLogic Server Console to verify JMS deployment.&lt;br&gt;<strong>Method</strong>&lt;br&gt;logJmstopicConnectionDown()</td>
</tr>
<tr>
<td>600108</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;An OMS Exception has been thrown. Reason: (0)&lt;br&gt;<strong>Method</strong>&lt;br&gt;logOMSException(Throwables th)</td>
</tr>
<tr>
<td>600109</td>
<td>debug</td>
<td><strong>Message Body</strong>&lt;br&gt;Lock successful&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;Exclusive access to events table granted&lt;br&gt;<strong>Method</strong>&lt;br&gt;logEventTableLockSuccess()</td>
</tr>
<tr>
<td>600110</td>
<td>debug</td>
<td><strong>Message Body</strong>&lt;br&gt;Lock failed&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;Access to events table denied&lt;br&gt;<strong>Cause</strong>&lt;br&gt;Other instance of Poller servlet holds the lock&lt;br&gt;<strong>Action</strong>&lt;br&gt;Wait for the current operation to end, and try again.&lt;br&gt;<strong>Method</strong>&lt;br&gt;logEventTableLockFailure()</td>
</tr>
</tbody>
</table>
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>600111</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;Invalid value ((1)) is specified for ((0)) in configuration file. Defaulting to ((2)).<strong>&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;Provided value for a parameter is missing or is invalid. A default value is found in oms-config-defaults.xml file and is defaulted to.</strong>&lt;br&gt;<strong>Cause</strong>&lt;br&gt;Value out of range - nonexisting path - invalid timezone id - incorrect boolean value**&lt;br&gt;<strong>Action</strong>&lt;br&gt;Check max and min values in oms-config-defaults.xml file - correct the path string - find the correct id for your timezone - check how boolean values are represented in oms-config-defaults.xml file**&lt;br&gt;<strong>Method</strong>&lt;br&gt;logInvalidValueForParameter(String property, String value, String default)</td>
</tr>
<tr>
<td>600112</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;Invalid value ((1)) is specified for ((0)) in configuration file and no default value found for it in oms-config-defaults.xml file.<strong>&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;Provided value for parameter is missing or is invalid. A default value is not found in oms-config-defaults.xml to default to.</strong>&lt;br&gt;<strong>Cause</strong>&lt;br&gt;oms-config-default.xml file cannot be accessed or it does not provide necessary information for the property in question**&lt;br&gt;<strong>Action</strong>&lt;br&gt;verify oms-config.xml and oms-config-defaults.xml files. check the installation. check the spelling of parameters in oms-config.xml file**&lt;br&gt;<strong>Method</strong>&lt;br&gt;logInvalidValueAndNoDefault(String property, String value)</td>
</tr>
<tr>
<td>600114</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;Error reading configuration parameters XML file(URL: [0]) - [1] - [2]<strong>&lt;br&gt;<strong>Message Details</strong>&lt;br&gt;An error happened while accessing or reading the indicated configuration file.</strong>&lt;br&gt;<strong>Cause</strong>&lt;br&gt;passing error - a malformed URL F119G120E120- G120I/O exception**&lt;br&gt;<strong>Method</strong>&lt;br&gt;logErrorLoadingConfigXMLFile(String url, String cause, String msg)</td>
</tr>
<tr>
<td>600115</td>
<td>error</td>
<td><strong>Message Body</strong>&lt;br&gt;In class [0], an SQL Statement was not closed after use. It will now be closed.<strong>&lt;br&gt;<strong>Cause</strong>&lt;br&gt;AbstractProxy.close() was not called after using SQL connection</strong>&lt;br&gt;<strong>Action</strong>&lt;br&gt;Report problem to development**&lt;br&gt;<strong>Method</strong>&lt;br&gt;logStatementNotClosed(String classname)</td>
</tr>
<tr>
<td>Error Code</td>
<td>Severity</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 600116     | error    | **Message Body**  
In class [0], an SQL Connection was not closed after use. It will now be closed.  
**Cause**  
AbstractProxy.close() was not called after using SQL connection  
**Action**  
Report problem to development  
**Method**  
logConnectionNotClosed(String classname) |
| 600117     | warning  | **Message Body**  
The view node [0] of type [1] has an invalid default value of [2]. The value has been ignored.  
**Cause**  
The default value is not entered correctly, or is incorrect for the data type.  
**Action**  
Change default value.  
**Method**  
logInvalidDefaultOrderNodeValue(String mnemonicPath, String nodeType, String default) |
| 600118     | error    | **Message Body**  
Unable to locate resource bundle.  
**Cause**  
Resource does not exist.  
**Action**  
Make sure there exist a resource file named [0]_[1].properties with key(s) [2].  
**Method**  
logI18NMissingResourceException(String baseName, String locale, String key) |
| 600119     | error    | **Message Body**  
Poller failed in init, message=[0].  
**Method**  
logPollerInitFailure(String msg) |
| 600120     | error    | **Message Body**  
Processing of timeout event failed, message=[0].  
**Method**  
logPollerProcessTimeoutFailed(String msg) |
| 600121     | error    | **Message Body**  
Unexpected system exception processing XML request: [0]  
**Cause**  
Unexpected system exception processing XML request: [0]  
**Action**  
Report to development  
**Method**  
logXMLAPIProcessorError(Throwable th) |
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600122     | debug    | Message Body
XMLAPI Servlet, message={0}.
Method
logXMLAPIServletTrace(String msg) |
| 600123     | error    | Message Body
Could not load XMLAPI Servlet request mappings. Servlet will be unavailable.
Cause
xml-request-mappings.xml file is missing.
Action
check web application WEB-INF/conf directory.
Method
logRequestMappingUnavailable(Exception reason) |
| 600127     | warning  | Message Body
Error reading XML Document: {0} |
| 600128     | warning  | Message Body
An error occurred processing XML request: {0} | {1} : message {2} |
| 600129     | warning  | Message Body
SQL Exception {0} : {1} |
| 600132     | error    | Message Body
Unable to add attachment. Attachment exceeds maximum file size={0} specified in the configuration file.
Message Details
Unable to add attachment. Attachment exceeds maximum file size={0} specified in the configuration file.
Method
logAddAttachmentExceedMaxFileSize(String maxFileSize) |
| 600133     | error    | Message Body
Invalid oms configuration on attachments. Attachment file system must not be null or empty. Maximum attachment size must be bigger than zero.
Message Details
Invalid oms configuration on attachments. Attachment file system must not be null or empty. Maximum attachment size must be bigger than zero.
Method
logInvalidOmsConfigurationOnAttachments() |
### Table 14–1 (Cont.) OSM Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 600134     | error    | **Message Body**
|            |          | Missing T3 file service with name={0}.  |
|            |          | **Message Details**
|            |          | Missing T3 file service with name={0}.  |
|            |          | **Cause**
|            |          | Missing configuration of T3 file server name in WebLogic console.  |
|            |          | **Action**
|            |          | Contact your local administrator to set T3 file server.  |
|            |          | **Method**
|            |          | logMissingT3FileServiceName(String fileName) |
| 600135     |          | **Message Body**
|            |          | T3 file service with name={0} must target exactly to one server.  |
|            |          | **Message Details**
|            |          | T3 file service with name={0} must target exactly to one server.  |
|            |          | **Cause**
|            |          | T3 file service was targeted to more than one server.  |
|            |          | **Action**
|            |          | Contact your local administrator to set up T3 file server properly.  |
|            |          | **Method**
|            |          | logDuplicateT3FileServiceTarget(String fileName) |
| 600136     | error    | **Message Body**
|            |          | Invalid T3 file service path.  |
|            |          | **Message Details**
|            |          | Invalid T3 file service path.  |
|            |          | **Cause**
|            |          | Invalid T3 file service path.  |
|            |          | **Action**
|            |          | Make sure T3 file service path is valid.  |
|            |          | **Method**
|            |          | logInvalidT3FileServicePath() |
| 600138     | error    | **Message Body**
|            |          | Targeted server = [{0}] for the T3 File Service with name = [{1}] is not running.  |
|            |          | **Message Details**
|            |          | Targeted server = [{0}] for the T3 File Service with name = [{1}] is not running.  |
|            |          | **Cause**
|            |          | Targeted server = [{0}] for the T3 File Service with name = [{1}] is not running.  |
|            |          | **Action**
|            |          | Make sure the targeted server is running.  |
|            |          | **Method**
|            |          | logTargetedServerForT3FileServiceNotRunning(String t3TargetServerName, String fileName) |
| 600139     | debug    | **Message Body**
|            |          | {0} |
# Automation Catalog Messages

Table 14–2 shows Automation Catalog messages.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>700002</td>
<td>debug</td>
<td>Message Body&lt;br&gt;parseXmlData(String xmlData)&lt;br&gt;Method&lt;br&gt;Parse the XML data [0]</td>
</tr>
<tr>
<td>700004</td>
<td>debug</td>
<td>Message Body&lt;br&gt;lookupOMSBean(String jndiName)&lt;br&gt;Method&lt;br&gt;Lookup the OMSSession Bean with jndiName=[0]</td>
</tr>
<tr>
<td>700007</td>
<td>debug</td>
<td>Message Body&lt;br&gt;receiveTask(long orderId, long orderHistoryId, String taskMnemonic)&lt;br&gt;Method&lt;br&gt;Receive Task with orderId=[0] orderHistoryId=[1] and task mnemonic=[2]</td>
</tr>
<tr>
<td>700008</td>
<td>debug</td>
<td>Message Body&lt;br&gt;completeTaskOnExit(long orderId, long orderHistId, String taskMnemonic, String comepletStat)&lt;br&gt;Method&lt;br&gt;Complete task On Exit with orderId=[0], orderHistoryId=[1], taskMnemonic=[2] and Completion Status=[3]</td>
</tr>
<tr>
<td>700009</td>
<td>debug</td>
<td>Message Body&lt;br&gt;assignTask(long orderId, long orderHistoryId, String taskMnemonic, String userId)&lt;br&gt;Method&lt;br&gt;Assign Task to [3] with orderId=[0] orderHistId=[1] and TaskMnemonic=[2]</td>
</tr>
<tr>
<td>700010</td>
<td>debug</td>
<td>Message Body&lt;br&gt;suspendTask(long orderId, long orderHistoryId, String taskMnemonic, String suspendState)&lt;br&gt;Method&lt;br&gt;SuspendTask to state=[3] with OrderID=[0], OrderHistId=[1] and Task Mnemonic=[2]</td>
</tr>
<tr>
<td>700011</td>
<td>debug</td>
<td>Message Body&lt;br&gt;acceptTask(long orderId, long orderHistoryId, String taskMnemonic)&lt;br&gt;Method&lt;br&gt;Accept Task with OrderId=[0] orderHistoryId=[1] and taskMnemonic=[2]</td>
</tr>
<tr>
<td>700012</td>
<td>debug</td>
<td>Message Body&lt;br&gt;updateOrderData1(long orderId, long orderHistId, String TaskMnem, String updateData)&lt;br&gt;Method&lt;br&gt;UpdateOrderData with orderId=[0] orderHistoryId=[1], TaskMnemonic=[2] and XML Data=[3]</td>
</tr>
</tbody>
</table>
## Table 14–2 (Cont.) Automation Catalog Messages

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Severity</th>
<th>Description</th>
</tr>
</thead>
</table>
| 700014     | debug    | Message Body  
The orderData id={0} and Data:{1}  
Method  
printOrderDataAsXML(long OrderId, String OrderData) |
| 700016     | debug    | Message Body  
Lookup the Automator: type={0} and jndiName={1}  
Method  
lookupAutomator(String type, String jndiName) |
| 700019     | debug    | Message Body  
Get a JMS Message with EventType={0}, Mnemonic={1}  
Method  
getAMessage(String type, String mnemonic) |
| 700020     | debug    | Message Body  
Set the outMessage Correlation to={0} the correlationId={1}  
Method  
setCorrelationId(String correlation, String correlationId) |
| 700021     | debug    | Message Body  
Send Notification Email message to :{0}, subject:{1}, MessageBody{2}  
Method  
sendEmailNotification(String to, String subject, String msgBody) |
| 700023     | error    | Message Body  
An automation exception has occurred At {0}, the reason is :{1}  
Cause  
See message body.  
Method  
AutomationException(String msg, Throwable th) |
| 700024     | error    | Message Body  
Naming Exception has occurred at {0} reason {1}  
Cause  
Cannot find user plug-in from JNDI tree.  
Method  
namingException(String name, Throwable ex) |
| 700033     | error    | Message Body  
{0} {1}  
Cause  
See message body for details.  
Method  
logAutomationException(String desc, Throwable exception) |