Oracle® Identity Manager

Installation and Configuration Guide for Oracle Application Server

Release 9.1.0

E10368-03

June 2008

ORACLE®

Oracle Identity Manager Installation and Configuration Guide for Oracle Application Server Release 9.1.0

E10368-03

Primary Author: Lyju Vadassery

Contributing Authors: Debapriya Datta, Shiladitya Guha

Contents

4 Installing and Configuring a Database for Oracle Identity Manager

5 Installing Oracle Identity Manager on Microsoft Windows

6 Installing Oracle Identity Manager Server on UNIX

7 Postinstallation Configuration for Oracle Identity Manager and Oracle Application Server

8 Starting and Stopping Oracle Identity Manager

9 Deploying in a Clustered Oracle Application Server Configuration

10 Installing and Configuring the Oracle Identity Manager Design Console

11 Installing and Configuring the Oracle Identity Manager Remote Manager

Preface

This guide explains the procedure to install Oracle Identity Manager release 9.1.0 on Oracle Application Server.

Audience

This guide is intended for system administrators of Oracle Identity Manager.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

http://www.oracle.com/accessibility/

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documents

For more information, see the following documents in the Oracle Identity Manager documentation set:

- *Oracle Identity Manager Release Notes*
- *Oracle Identity Manager Installation and Configuration Guide for JBoss Application Server*
- *Oracle Identity Manager Installation and Configuration Guide for BEA WebLogic Server*
- *Oracle Identity Manager Installation and Configuration Guide for IBM WebSphere Application Server*
- *Oracle Identity Manager Best Practices Guide*
- *Oracle Identity Manager Globalization Guide*
- *Oracle Identity Manager Design Console Guide*
- *Oracle Identity Manager Administrative and User Console Guide*
- *Oracle Identity Manager Administrative and User Console Customization Guide*
- *Oracle Identity Manager Tools Reference*
- *Oracle Identity Manager Audit Report Developer's Guide*
- *Oracle Identity Manager Integration Guide for Crystal Reports*
- *Oracle Identity Manager API Usage Guide*
- *Oracle Identity Manager Concepts*
- *Oracle Identity Manager Reference*

Documentation Updates

Oracle is committed to delivering the best and most recent information available. For information about updates to the Oracle Identity Manager documentation set, visit Oracle Technology Network at

`http://www.oracle.com/technology/documentation`

Conventions

The following text conventions are used in this document:

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

Convention	Meaning
**_HOME*	This convention represents the directory where an application is installed. The directory where you install Oracle Identity Manager is referred to as *OIM_HOME*. Each Oracle Identity Manager component includes an abbreviation: *OIM_DC_HOME* for the Design Console and *OIM_RM_HOME* for the Remote Manager. *ORACLE_HOME* represents the directory where Oracle Application Server is installed.
OC4J	Oracle Containers for J2EE

<Entry 1>.<Entry 2>.<Entry 3>	This convention represents nested XML entries that appear in files as follows:
	
	`<Entry 1>`
	`  <Entry 2>`
	`  <Entry 3>`

1

Overview of the Installation Procedure

Installing Oracle Identity Manager on Oracle Application Server involves the following steps:

1. Preparing for the installation: See Chapter 2, "Planning the Installation".
2. Setting up Oracle Application Server for Oracle Identity Manager: See Chapter 3, "Installing and Configuring Oracle Application Server for Oracle Identity Manager".
3. Setting up a database for Oracle Identity Manager: See Chapter 4, "Installing and Configuring a Database for Oracle Identity Manager".
4. Installing a single Oracle Identity Manager instance: See one of the following chapters based on the operating system:
   - Chapter 5, "Installing Oracle Identity Manager on Microsoft Windows"
   - Chapter 6, "Installing Oracle Identity Manager Server on UNIX"
5. Performing basic Oracle Identity Manager Server and Oracle Application Server configuration tasks related to the installation setup: See Chapter 7, "Postinstallation Configuration for Oracle Identity Manager and Oracle Application Server".
6. Starting Oracle Identity Manager and accessing the Administrative and User Console: See Chapter 8, "Starting and Stopping Oracle Identity Manager".
7. Installing and configuring Oracle Identity Manager in a clustered Oracle Application Server environment: See Chapter 9, "Deploying in a Clustered Oracle Application Server Configuration".
8. Installing, configuring, and starting the Oracle Identity Manager Design Console: See Chapter 10, "Installing and Configuring the Oracle Identity Manager Design Console".
9. Installing, configuring, and starting the Oracle Identity Manager Remote Manager: See Chapter 11, "Installing and Configuring the Oracle Identity Manager Remote Manager".
10. Troubleshooting the Oracle Identity Manager installation: See Chapter 12, "Troubleshooting the Oracle Identity Manager Installation".

2

Planning the Installation

Oracle recommends that you familiarize yourself with the components required for deployment before installing Oracle Identity Manager. Oracle also recommends that you install and use the included Diagnostic Dashboard to ensure that your system is ready for Oracle Identity Manager installation. Refer to the "Using the Diagnostic Dashboard" section on page 2-4 for details of installing the Diagnostic Dashboard.

A basic Oracle Identity Manager installation consists of the following:

- A database server
- An application server
- An Oracle Identity Manager server running on the application server
- A Design Console
- An Administrative and User Console running on a Web browser

This chapter contains the following topics:

Host Requirements for Oracle Identity Manager Components

This section lists the minimum host system requirements for the various components in an Oracle Identity Manager environment.

> **Note:** Check the Oracle Identity Manager Release Notes for the requirements and supported configurations specific to each version of the Oracle Identity Manager product.

You must obtain the enterprise versions of the application server and database software, complete with valid licenses. Oracle Identity Manager does not include this software.

The Oracle Identity Manager installation program might conflict with other installed applications, utilities, or drivers. Try to remove all nonessential software and drivers from the installation computer before loading Oracle Identity Manager. This practice also ensures that the database host can create the database schema.

Oracle Identity Manager Server Host Requirements

Table 2–1 lists the minimum host requirements for Oracle Identity Manager server and the guidelines for a basic installation.

Table 2–1 Oracle Identity Manager Server Host Requirements

Server Platform	Item
Microsoft Windows and Linux	▪ Processor Type: Intel Xeon or Pentium IV ▪ Processor Speed: 2.4 GHz or higher, 400 MHz FSB or higher ▪ Number of Processors: 1 ▪ Memory: 2 GB for each Oracle Identity Manager Server instance ▪ Hard Disk Space: 1 GB (initial size)
Solaris	▪ Server: Sun Fire V210 ▪ Number of Processors: 1 ▪ Memory: 2 GB for each Oracle Identity Manager Server instance ▪ Hard Disk Space: 1 GB (initial size)

Database Server Host Requirements

Table 2–2 provides sample database minimum host requirements for selective supported operating systems and should be considered only as guidelines. Refer to database documentation for the specific database host requirements.

Table 2–2 Sample Database Server Host Requirement

Database Server Platform	Item
Microsoft Windows and Linux	▪ Processor Type: Intel Xeon ▪ Processor Speed: 2.4 GHz or higher, 400 MHz FSB or higher ▪ Number of Processors: 2 ▪ Memory: 4 GB total or 2 GB for each CPU ▪ Hard Disk Space: 40 GB (initial size) for Microsoft Windows, 20 GB (initial size) for UNIX
Solaris	▪ Server: Sun Fire V250 ▪ Number of Processors: 2 ▪ Memory: 4 GB total or 2 GB for each CPU ▪ Hard Disk Space: 40 GB (initial size) ▪ Number of Hard Disks: 1 Disk

Design Console Host Requirements

Table 2–3 lists the minimum host requirements for the Oracle Identity Manager Design Console.

Table 2–3 Design Console Host Requirements

Design Console Platform	Item
Microsoft Windows	■ Processor Type: Intel Pentium IV ■ Processor Speed: 1.4 GHz or higher ■ Number of Processors: 1 ■ Memory: 512 MB ■ Hard Disk Space: 300 MB

Remote Manager Host Requirements

Table 2–4 lists the minimum host requirements for the Oracle Identity Manager Remote Manager.

Table 2–4 Remote Manager Host Requirements

Remote Manager Platform	Item
Microsoft Windows and Linux	■ Processor Type: Intel Pentium IV ■ Processor Speed: 1.4 GHz or higher ■ Number of Processors: 1 ■ Memory: 512 MB ■ Hard Disk Space: 1 GB
Solaris	■ Sun Fire V100 Server ■ Number of Processors: 1 ■ Memory: 512 MB ■ Hard Disk Space: 10 GB
AIX	■ Processor Type: PowerPC ■ Number of Processors: 1 ■ Memory: 512 MB ■ Hard Disk Space: 10 GB

Planning for Non-English Oracle Identity Manager Environments

If you are deploying Oracle Identity Manager components in non-English environments, then review the following guidelines and requirements:

- Before installing any of the Oracle Identity Manager components, ensure that the regional and language settings (locale) on the target system meet the following requirements:
  - An appropriate language version of the operating system is installed.
  - Specific language settings are properly configured.
- Refer to *Oracle Identity Manager Globalization Guide* for information about configuring localized deployments and to ensure you meet the character restrictions for various components and attributes.
- For Oracle database globalization support, you must configure the database for Unicode. Refer to "Creating an Oracle Database" on page 4-1 for more information.

Installation Worksheet

Table 2–5 provides information about the configuration attributes that you must set during Oracle Identity Manager installation. Print this worksheet and use it to take notes during the installation. Enter information specific to your installation in the User Selection column.

Table 2–5 Installation Worksheet

Item	Default	User Selection
The base directory for installing Oracle Identity Manager.	Microsoft Windows: C:\oracle UNIX: /opt/oracle	
The name or IP address of the computer where the Oracle Identity Manager database is installed.	Not Applicable for a default. However you must enter a value for this item when you install Oracle Identity Manager	
The TCP port number on which the database listens for connections.	1521 for Oracle	
The name of the database for your installation.	No default value	
The name and password of the database account that Oracle Identity Manager uses to access the database.	No default value	
The JDK installation directory	Microsoft Windows: *ORACLE_HOME*\jdk UNIX: *ORACLE_HOME*/jdk	
The Oracle Application Server installation directory	Microsoft Windows: C:\product\10.1.3.1\OracleAS_1 UNIX: /opt/product/10.1.3.1/OracleAS_1	

Using the Diagnostic Dashboard

The Diagnostic Dashboard is a Web application that runs in the application server. It checks your pre and postinstallation environments for components required by Oracle Identity Manager. Oracle recommends that you install the Diagnostic Dashboard before installing Oracle Identity Manager.

Installing the Diagnostic Dashboard

The Diagnostic Dashboard files are located in the `DiagnosticDashboard directory` on the Oracle Identity Manager Installer CD media.

You must deploy the Diagnostic Dashboard Web application on your application server.

> **See Also:** *Oracle Identity Manager Administrative and User Console Guide* for more information about the Diagnostic Dashboard

Verifying Your Preinstallation Environment

You can use the Diagnostic Dashboard to verify that the components required to install Oracle Identity Manager are present:

- A supported Java Virtual Machine (JVM)
- A supported database

See Also: *Oracle Identity Manager Administrative and User Console Guide* for information about the Diagnostic Dashboard

3

Installing and Configuring Oracle Application Server for Oracle Identity Manager

This chapter explains how to set up Oracle Application Server before installing Oracle Identity Manager. You must perform the following tasks described in this chapter:

> **Note:** Follow the instructions in Chapter 9, "Deploying in a Clustered Oracle Application Server Configuration" if you are deploying Oracle Identity Manager on clustered Oracle Application Server.

- Installing Oracle Application Server
- Upgrading Oracle Application Server from Version 10.1.3.1 to Version 10.1.3.3
- Creating OC4J Instance
- Applying Oracle Application Server Patches
- Setting the RMI Port Number Range
- Setting Environment Variables
- Verifying the Java JDK Version
- Creating a Backup of the Oracle Application Server Configuration

Installing Oracle Application Server

When you run the Oracle SOA Suite installer program, you must choose the **Advanced Install** option and choose the **J2EE Server and Web Server** option on the Select Installation Type page.

Ensure that you select **Configure this as an Administrator OC4J instance** option to install the administrative application on the server.

After the installer finishes, the OC4J instance within your Oracle Application Server instance starts automatically.

Change the shell limits specific to the operating system as specified in the installation guides to ensure stable performance of Oracle Identity Manager.

Note: Oracle strongly recommends that you must change the shell limits specific to your operating system to resolve many Oracle Identity Manager runtime issues. For more information, refer to the "Requirements" section of the Oracle Application Server Installation Guide for 10*g*Release 3 (10.1.3.1.0) for your operating system.

Upgrading Oracle Application Server from Version 10.1.3.1 to Version 10.1.3.3

Ensure that you have upgraded Oracle Application Server from version 10.1.3.1 to version 10.1.3.3.

To check the version of Oracle Application Server that you currently use:

1. Ensure that Oracle Application Server's Java binary (*ORACLE_HOME*/jdk/bin) is in PATH.
2. Run the following command from *ORACLE_HOME*\j2ee*OC4J_INSTANCE*\java -jar oc4j.jar version.

   The information that appears should be similar to the following example:

   ```
   Oracle Containers for J2EE 10g (10.1.3.3.0) (build
   070610.1800.23513)
   ```

Creating OC4J Instance

Oracle recommends that you create an OC4J instance for installing Oracle Identity Manager, so that the default *home* OC4J instance can be used only for hosting administrative application.

This makes it possible for restarting the instance where the Oracle Identity Manager is deployed using the administrative console. Note that the administrative console application is deployed to the default home instance.

To create an OC4J instance:

1. Log in to the administrative application server console, click the name of the application server, and then click **Create OC4J Instance**.
2. Enter the OC4J instance name, select **Add to a new group with name**, and then enter a new group name.

Applying Oracle Application Server Patches

The following patches may be specific to 10.1.3.3.0 version of Oracle Application Server:

Note:

- If you are running Oracle Identity Manager on any other version of Oracle Application Server, then contact Oracle to get the patches for that version of Oracle Application Server. Patches and instructions on how to apply those patches can be downloaded from the Oracle*MetaLink* Web site at:

  http://metalink.oracle.com.

- If you are installing Oracle Identity Manager in Microsoft Windows Vista, then you must set the environment variable `OPATCH_PLATFORM_ID` to `207`.

1. Install OPatch patch 2617419.
2. Install Oracle Application Server patch 6685235.
3. Install Oracle Application Server patch 5389650.
4. Install Oracle Application Server patch 6454278.
5. Open *ORACLE_HOME*`/j2ee/`*OC4J_INSTANCE*`/config/rmi.xml`. Add max-server-sockets="200" as below:

```
<rmi-server
...
max-server-sockets="200"
>
```

Note: *OC4J_INSTANCE* is the instance of Oracle Application Server on which Oracle Identity Manager is deployed.

Setting the RMI Port Number Range

The Oracle Process Manager and Notification server (OPMN) dynamically assigns port numbers to each OC4J instance within your Oracle Application Server instance. To ensure you can access the Oracle Identity Manager Design Console and Administrative and User Console on Oracle Application Server, you must set the RMI port number range to be unique in the *ORACLE_HOME*`/opmn/conf/opmn.xml` file. Perform the following steps to set a unique RMI port number range:

1. Open the *ORACLE_HOME*`/opmn/conf/opmn.xml` file with a text editor.
2. Locate the `<port id="rmi" range="12401-12500"/>` entry for the instance you will install Oracle identity Manager on.
3. Choose one of the ports for RMI within the range of 12401 and 12500.

   For example:

```
<port id="rmi" range="12408"/>
```

Note: When you install Oracle Identity Manager on the installer's Application Server Information page, you must enter the RMI port number you set in the opmn.xml file in the **RMI Port No** field.

4. Save and close the opmn.xml file and restart the OC4J instance.

For Oracle Application Server clusters, repeat these steps on each node in the cluster so that each opmn.xml file contains the same unique port number.

Setting Environment Variables

After you have verified that Oracle Identity Manager is using the JDK included with Oracle Application Server (refer to "Verifying the Java JDK Version" for more information), complete the following steps to set your environment variables:

For Microsoft Windows

1. From the Windows **Start Menu**, select **Settings**, **Control Panel**, **System**, **Advanced**, and then select **Environment Variables**.

   In the scroll box labeled **System Variables**, select **Path**, then click **Edit**.

   In the Variable Value field, add the location of your JDK to the beginning of the existing path. For example, if your existing path is the following:

   ```
   %SystemRoot%\system32;%SystemRoot%;C:\Program Files;
   ```

   Change it to the following:

   ```
   ORACLE_HOME\jdk\bin;%SystemRoot%\system32;%SystemRoot%;C:\Program Files
   ```

   Click **OK** to commit your change.

2. In the **System Variables** list, search for JAVA_HOME.

   If JAVA_HOME does not exist, complete Step a. If JAVA_HOME exists, complete Step b.

   a. Click **New**. In the **Variable Name** field, enter `JAVA_HOME`. In the field labeled **Variable Value** field, enter the path to the JDK, for example: *ORACLE_HOME*`\jdk`.

      Click **OK** to commit your entries, then click **OK** twice more to close the Environment Variables and System Properties windows, respectively.

   b. Click **Edit**. Verify that the path to the JDK exists in the field labeled **Variable Value** field. If it does not exist, enter the path in the **Variable Value** field, for example: *ORACLE_HOME*`\jdk`.

      Click **OK** to commit your entry, then click **OK** twice more to close the Environment Variables and System Properties windows, respectively.

      Note: A message might appear prompting you to update the JDK. Close this window **without** updating the JDK, because Oracle Identity Manager Release 9.1.0 for Oracle Application Server requires that you use the JDK included with Oracle Application Server.

For UNIX

1. To set the JAVA_HOME variable, run the following command:

   ```
   export JAVA_HOME=ORACLE_HOME/jdk
   ```

2. To set the Java binary in the PATH variable, run the following command:

   ```
   export PATH=ORACLE_HOME/jdk/bin:$PATH
   ```

Verifying the Java JDK Version

Oracle Identity Manager for Oracle Application Server requires you to use the JDK included with Oracle Application Server. Remove any other JDK versions from your system.

The following procedure explains how to verify that the correct version of the Java JDK is used by Oracle Identity Manager. To verify on a Microsoft Windows system:

1. Open a console window.
2. Enter `java -version`

   For example, the information that appears should look like the following:

```
C:\>java -version
java version "1.5.0_06"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_06-b05)
Java HotSpot(TM) Client VM (build 1.5.0_06-b05, mixed mode)
```

Creating a Backup of the Oracle Application Server Configuration

It is recommended that you create a backup of the Oracle Application Server at regular intervals during the process of Oracle Identity Manager installation. Creating a backup makes it possible to rollback the changes if you run into any issues.

Preparing Oracle Application Server for Backup

You must prepare the Oracle Application Server before you create a backup of the configuration. This is to be performed only once. You do not need to perform this procedure each time you create a backup of the configuration.

To prepare the Oracle Application Server configuration for backup:

1. Create a directory named `backupfiles` under the *ORACLE_HOME*`/backup_restore/` directory.
2. Edit *ORACLE_HOME*`/backup_restore/config/config.inp` and change all instances of `VALUE_NOT_SET` to *ORACLE_HOME*`/backup_restore/backupfiles` under the `REQUIRED PARAMETERS` section only.
3. For Microsoft Windows, open a command prompt and run:

   ```
   set ORACLE_HOME=ORACLE_HOME
   ```

   For UNIX, open shell and run:

   ```
   export ORACLE_HOME=ORACLE_HOME
   ```

4. Go to the *ORACLE_HOME*`/backup_restore/` directory and run:

   ```
   bkp_restore.bat -m configure
   ```

Creating a Backup of the Configuration

To create a backup of the Oracle Application Server configuration, run the following script from the *ORACLE_HOME*`/backup_restore/` directory:

```
bkp_restore.bat -m backup_instance_cold
```

Restoring the Configuration

If you want to restore the Oracle Application Server configuration, then run the following script from the *ORACLE_HOME*/backup_restore/ directory:

```
bkp_restore.bat -m restore_instance
```

This will list out all the available timings. Then, you can use the -t option to restore the configuration. For example:

```
bkp_restore.bat -m restore_instance -t 2006-09-21_06-12-45
```

Note: The instructions to restore the configuration are given for reference only. For more information about creating a backup of the Oracle Application Server configuration, see "Chapter 17: Backup Strategy and Procedures" in the Oracle Application Server Administrator's Guide.

4

Installing and Configuring a Database for Oracle Identity Manager

Oracle Identity Manager requires a database. You must install and configure your database before you begin the Oracle Identity Manager installation. Refer to the topic that applies to your database:

- Using an Oracle Database for Oracle Identity Manager
- Using Oracle RAC Databases for Oracle Identity Manager

Using an Oracle Database for Oracle Identity Manager

To use Oracle Database as your database, you must perform the tasks described in the following sections:

- Installing Oracle Database
- Creating an Oracle Database
- Preparing the Oracle Database

Installing Oracle Database

Install Oracle9*i* Database or Oracle Database 10*g* release 2 by referring to the documentation delivered with the Oracle database. See *Oracle Identity Manager Release Notes* for the specific supported versions. Oracle recommends using the Basic installation.

> **Note:** If you choose the Custom installation, you must include the JVM option, which is required for XA transaction support.

Creating an Oracle Database

You can create a new Oracle database instance for Oracle Identity Manager. When creating the database, ensure that you configure the Oracle JVM feature and enable query rewrite.

You can use the Database Configuration Assistant (DBCA) tool to create the database. To configure the Oracle JVM feature, select the Oracle JVM feature on the Standard Database Features page of the DBCA.

To enable the database for query rewrite, set the initialization parameters `QUERY_REWRITE_ENABLED` to `TRUE` and `QUERY_REWRITE_INTEGRITY` to `TRUSTED` in the **All Initialization Parameters** field of the DBCA.

> **Note:** For the Oracle Identity Manager installation, Oracle recommends that you configure a minimum block size of 8K for Oracle Database.

Refer to your Oracle Database documentation for detailed instructions on creating a database instance.

Configuring the Database for Globalization Support

For globalization support for Oracle Identity Manager, Oracle recommends that for configuring the database for Unicode. To configure the database for Unicode, perform the following steps:

1. Select **AL32UTF8** in the Character Sets tab of the DBCA. This character set supports the Unicode standard.
2. Set the `NLS_LENGTH_SEMANTICS` initialization parameter to `CHAR` in the **All Initialization Parameters** field of the DBCA.

> **See Also:** *Oracle Identity Manager Globalization Guide* for information about globalization support for Oracle Identity Manager

Preparing the Oracle Database

After you install Oracle Database and created a database instance, you must prepare it for Oracle Identity Manager by completing the following tasks:

- Verify that query rewrite is enabled

> **Note:** Query rewrite is applicable only if you are using Oracle Database Enterprise Edition.

- Enable XA transactions support

> **Note:** Java Virtual Machine (JVM) is required to enable XA transaction support. If you did not install the Oracle JVM component during Oracle Database installation, then you must install it now. See the Oracle Database documentation for specific instructions.

- Create at least one tablespace for storing Oracle Identity Manager data
- Create a database user account for Oracle Identity Manager

You can perform the preceding tasks to prepare your Oracle database for Oracle Identity Manager by running one of the following scripts:

- On UNIX, run the following:

  ```
  prepare_xl_db.sh
  ```

- On Microsoft Windows, run the following:

  ```
  prepare_xl_db.bat
  ```

Both of these scripts ship with the Oracle Identity Manager Installer and are located in the `/installServer/Xellerate/db/oracle/` directory.

You must observe the following prerequisites when using the prepare_xl_db script:

- The script must be run by a user holding DBA privileges (for example, the oracle user on UNIX typically holds these privileges).
- The script must be run on the computer on which the database resides.

To prepare your Oracle database for Oracle Identity Manager, complete the steps associated with the operating system on the computer hosting your Oracle database:

Preparing the Database on UNIX:

To prepare the database on UNIX:

1. Copy the scripts `prepare_xl_db.sh` and `xell_db_prepare.sql` from the distribution CD to a directory on the computer hosting your database on which you (as the account user performing this task) have write permission.
2. Run the following command to enable permission to run the script:

   ```
   chmod 755 prepare_xl_db.sh
   ```

3. Run the `prepare_xl_db.sh` script by entering the following command:

   ```
   ./prepare_xl_db.sh
   ```

4. Provide information appropriate for your database and host computer when the script prompts you for the following items:
   - The location of your Oracle home (*ORACLE_HOME*)
   - The name of your database (*ORACLE_SID*)
   - The name of the Oracle Identity Manager database user to be created
   - The password for the Oracle Identity Manager database user
   - The name of the tablespace to be created for storing Oracle Identity Manager data
   - The directory in which to store the data file for the Oracle Identity Manager tablespace
   - The name of the data file (you do not append the .dbf extension)
   - The name of the temporary tablespace
5. Check the `prepare_xl_db.lst` log file located in the directory from which you ran `prepare_xl_db.sh` to see the execution status and additional information.

Note: If you encounter errors after running the `prepare_xl_db.sh` script, run the following command to ensure that `prepare_xl_db.sh` is executable on UNIX and Linux and then run the `prepare_xl_db.sh` script again.

```
$ dos2unix prepare_xl_db.sh
```

Preparing the Database on Microsoft Windows:

To prepare the database on Microsoft Windows:

1. Copy the scripts `prepare_xl_db.bat` and `xell_db_prepare.sql` from the distribution CD to a directory on the computer hosting your database on which you (as the account user performing this task) have write permission.

2. Open a command window, navigate to the directory to which you just copied the scripts, then run `prepare_xl_db.bat` with the following arguments:

```
prepare_xl_db.bat ORACLE_SID ORACLE_HOME
XELL_USER XELL_USER_PWD TABLESPACE_NAME
DATAFILE_DIRECTORY DATAFILE_NAME
XELL_USER_TEMP_TABLESPACE SYS_USER_PASSWORD
```

For example, the string you enter on the command line might look like the following:

```
prepare_xl_db.bat XELL C:\oracle\ora92 xladm xladm
xeltbs C:\oracle\oradata xeltbs_01 TEMP manager
```

Table 4–1 lists the options used in the preceding example of `prepare_xl_db.bat`.

Table 4–1 Options for the prepare_xl_db.bat Script

Argument	Description
`XELL`	Name of the database
`C:\oracle\ora92`	Directory where the Oracle database is installed
`xladm`	Name of the Oracle Identity Manager user to be created
`xladm`	Password for the Oracle Identity Manager user
`xeltbs`	Name of the tablespace to be created
`C:\oracle\oradata`	Directory where the data files will be placed
`xeltbs_01`	Name of the data file (you do not need to include the .dbf extension)
`TEMP`	Name of the temporary tablespace that already exists in your database
`manager`	Password for the SYS user

3. Check the `prepare_xl_db.lst` log file located in the directory from which you ran prepare_xl_db.bat to see the execution status and additional information.

Evaluating Script Results

If the script returns a message indicating successful execution, you can continue to the next task, which is installing Oracle Identity Manager.

If the script does not succeed, you must manually fix all fatal (nonrecoverable) errors so that the database is prepared successfully.

You can ignore all non-fatal errors. For example, when the script tries to drop a non-existent view, it will return the error "ORA-00942: table or view does not exist".

Make sure to scan all the errors in the log file and ignore or resolve them on an individual basis. Remember that you must successfully prepare the database for Oracle Identity Manager before you can install Oracle Identity Manager.

Removing Oracle Identity Manager Entries from an Oracle Database

To remove Oracle Identity Manager entries from an Oracle database after removing (deinstalling) the Oracle Identity Manager product, drop the database user holding the Oracle Identity Manager schema.

Using Oracle RAC Databases for Oracle Identity Manager

This section explains how to deploy Oracle Real Application Clusters (Oracle RAC) databases for Oracle Identity Manager and contains the following sections:

Installing Oracle Identity Manager for Oracle RAC

Oracle RAC is a cluster database with a shared cache architecture that provides highly scalable and available database solutions. Oracle RAC consists of multiple database instances on different computers acting in tandem to provide these features.

Note: The Oracle Identity Manager Installer program does not provide support for Oracle RAC. To deploy Oracle Identity Manager for Oracle RAC, you must install Oracle Identity Manager on a single database instance in Oracle RAC and then change the application server settings, specifically the connection pool parameters, to use the Oracle RAC JDBC connection string.

Use the following steps to install Oracle Identity Manager for Oracle RAC:

1. Ensure that Oracle RAC is properly set up and configured with the Oracle Identity Manager schema owner.
2. Start the Oracle Identity Manager Installer.
3. On the Database Parameters page of the installer, enter the host name, port number, and database name of a single database instance in Oracle RAC.
4. Complete the Oracle Identity Manager installation by performing the steps in the installer.
5. Configure your application server for Oracle RAC by referring to Configuring Oracle Application Server for Oracle RAC.

Oracle RAC Net Services

The net services name entry for an Oracle RAC database differs from that of a conventional database. The following is an example of the net services name entry for an Oracle RAC database:

```
racdb=
      (DESCRIPTION=
              (LOAD_BALANCE=off)
              (FAILOVER=on)
              (ADDRESS_LIST=
                     (ADDRESS=(protocol=tcp)(host=node1-vip)(port=1521))
                     (ADDRESS=(protocol=tcp)(host=node2-vip)(port=1521)))
      (CONNECT_DATA=
              (SERVER=DEDICATED)
              (SERVICE_NAME=racdb)))
```

Table 4–2 lists and describes the parameters in a net services name entry for an Oracle RAC database.

Table 4–2 Parameters for Oracle RAC Database Net Services Name Entries

Parameter	Description
LOAD_BALANCE	Specifies whether client load balancing is enabled (on) or disabled (off). The default setting is on.
FAILOVER	Specifies whether failover is enabled (on) or disabled (off). The default setting is on.
ADDRESS_LIST	Specifies the list of all the nodes in Oracle RAC, including their host names and the ports they listen on.

JDBC and Oracle RAC

JDBC client applications using the Thin driver to connect to an Oracle RAC database must use the Oracle RAC net services name as a part of the JDBC URL. The entire Oracle RAC net services name is concatenated and the entire string is used in the JDBC URL so the client application can connect to Oracle RAC.

The following sample code shows how a JDBC URL is used to connect to an Oracle RAC database:

```
//String url = "jdbc:oracle:thin:@dbhost:1521:dbservice"
String racUrl =
"jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(ADDRESS_LIST=(ADDR
ESS=(protocol=tcp)(host=node1-vip)(port=1521))(ADDRESS=(protocol=tcp)(host=node2-v
ip)(port=1521)))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=racdb)))";

       String strUser = "username";
       String strPW = "password";

       // load Oracle driver
       Class.forName("oracle.jdbc.driver.OracleDriver");

       // create the connection
       con = DriverManager.getConnection(strURL, strUser, strPW);
```

The subsequent sections about configuring application servers for Oracle RAC databases explain how to modify connection pools to use a similar JDBC URL so the application server can communicate with Oracle RAC.

Configuring Oracle Application Server for Oracle RAC

This section explains how to configure Oracle Application Server (nonclustered or clustered) for Oracle RAC by ensuring the data sources and connection pools are configured to use the Oracle RAC JDBC connection string.

Note: Before configuring Oracle Application Server for Oracle RAC, you must:

- Get the RAC net services name from the tnsnames.ora file.
- Construct the RAC JDBC URL by referring to JDBC and Oracle RAC.

Perform the following steps to configure both non-clustered and clustered Oracle Application Servers for Oracle RAC:

> **Note:** If you are configuring an Oracle Application Server cluster for Oracle RAC, perform each of the following steps on all nodes in the cluster.

1. Open the *OIM_HOME*/xellerate/config/xlconfig.xml file.
2. Locate the <DirectDB> section and replace the value of the <url>...</url> tag with the RAC JDBC URL. For example, the new tag might be similar to the following:

   ```
   <url>jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=off)(FAILOVER=on)(ADDRESS_
   LIST=(ADDRESS=(protocol=tcp)(host=node1-vip)(port=1521))(ADDRESS=(protocol=tcp)
   (host=node2-vip)(port=1521)))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_
   NAME=racdb)))</url>
   ```

3. Save and close the *OIM_HOME*/xellerate/config/xlconfig.xml file.
4. Log in to the Oracle Application Server Administrative Console by using a Web browser.
5. Select the application server on which Oracle Identity Manager is installed and then select the OC4J instance within the Oracle Application Server instance you are configuring for Oracle RAC.
6. Select the **Administration** tab, then select **Services**, and then select **JDBC Resources**.
7. Locate the **Connection Pools** section and select **xlConnectionPool**.
8. Set the **URL** property value to the RAC JDBC URL described in step 2.
9. Save the settings.
10. Select **xlXAConnectionPool**.
11. Set the **URL** property value to the RAC JDBC URL described in step 2.
12. Save the settings.

> **Note:** For a clustered Oracle Application Server environment, repeat steps 5-12 for each node in the cluster.

13. Restart the Oracle Application Server. If you are configuring an Oracle Application Server cluster for Oracle RAC, restart all nodes in the cluster.

5

Installing Oracle Identity Manager on Microsoft Windows

This chapter explains how to install Oracle Identity Manager on Microsoft Windows in a nonclustered installation.

> **See Also:** Chapter 9, "Deploying in a Clustered Oracle Application Server Configuration" for information about deploying Oracle Identity Manager in a clustered installation

You must install Oracle Identity Manager on systems running the application server. Oracle Identity Manager components such as the Remote Manager and Design Console can be installed on separate systems. Each component has its own installer.

This chapter contains the following topics:

> **Caution:** Do *not* use a remote client tool, such as Symantec pcAnywhere, to install Oracle Identity Manager products.

Installing the Database Schema

As part of the installation, the Oracle Identity Manager Installer loads a schema into your database. You only install the database schema once. It is installed the first time you run the Oracle Identity Manager Installer. Each subsequent time you run the installer to deploy other Oracle Identity Manager components you enter information about the database connection to configure the component for the same schema. If required, contact your database administrator (DBA).

> **Note:** During the schema installation, a log file is created in the *OIM_HOME*\logs directory.

Installing Documentation

The Oracle Identity Manager documentation is installed automatically in the *OIM_HOME* directory. No special input is required. A full documentation set is installed with each Oracle Identity Manager component.

Installing Oracle Identity Manager on Microsoft Windows

This section describes how to install Oracle Identity Manager on a computer running Microsoft Windows.

Caution: Do not install Oracle Identity Manager on top of an existing Oracle Identity Manager installation. For each new installation, use a different home directory. If you want to reuse the name of an existing Oracle Identity Manager home directory, then backup your original Oracle Identity Manager home by renaming that directory.

Remember at all times that all Oracle Identity Manager components must be installed in different home directories. For example, you cannot install the Remote Manager in the same directory as Oracle Identity Manager.

To install Oracle Identity Manager on a Microsoft Windows host:

Note: Create a backup of the Oracle Application Server configuration before installing Oracle Identity Manager. For more information, see "Creating a Backup of the Oracle Application Server Configuration" on page 3-5.

---

1. Insert the Oracle Identity Manager Installation CD into your CD-ROM drive.
2. Using Windows Explorer, access the installServer directory on the installation CD and double-click the setup_server.exe file.
3. Select a language on the Installer page and click **OK**. The Welcome page is displayed.
4. Click **Next** on the Welcome page. The Admin User Information page is displayed.
5. Enter the password that you want to use as the Oracle Identity Manager administrator, confirm the password by entering it again, and then click **Next**. The OIM Application Options page is displayed.
6. Select one of the following applications to install, and then click **Next**:
   - Oracle Identity Manager
   - Oracle Identity Manager with Audit and Compliance Module

     See Also: *Oracle Identity Manager Audit Report Developer's Guide* for information about the Audit and Compliance Module

   The Target directory page is displayed.
7. Complete one of the following:
   - Install Oracle Identity Manager in the default directory, which is C:\oracle\, click **Next**.

- Install Oracle Identity Manager in another directory, enter the path in the **Directory** field, then click **Next**.

  or

  Click **Browse**, navigate to the desired location, then click **Next**.

  Note: If the directory path does not exist, then the Base Directory settings field is displayed. Click **OK**. This directory is automatically created. If you do not have write permission to create the default directory, then a message is displayed informing you that the installer could not create the directory. Click **OK** to close the message, then contact your system administrator to obtain the appropriate permissions.

8. On the Database Server Selection page, specify **Oracle** as the database that you are using with Oracle Identity Manager and click **Next**.

   Note: Only Oracle database is supported for Oracle Identity manager installation on Oracle Application Server.

9. On the Database Information page, provide all database connectivity information required to install the database schema. You install this schema just once, as part of your initial Oracle Identity Manager installation. Thereafter, you configure all the other Oracle Identity Manager components to point to this common schema.

   Note: To install against an existing database, verify that the version of Oracle Identity Manager you are installing is certified with your existing database version. See *Oracle Identity Manager Release Notes* for information about the certified configurations.

   When Oracle Identity Manager is installed against an existing database, a warning message is displayed indicating the database schema already exists and instructing you to copy the .xldatabasekey file from the existing Oracle Identity Manager installation to the new *OIM_HOME*\xellerate\config\ directory after you complete the installation process.

   You should create the \config directory in the new *OIM_HOME*\xellerate\ path if it does not already exist.

   Enter the following database information:

   - In the **host** field, enter the host name or the IP address of the computer on which the database resides.
   - In the **PORT** field, enter the port number on which the database listens for connections. The default port is 1521 for Oracle Database.
   - In **Database SID** field, enter the name of the database instance.
   - In the **User Name** field, enter the user name of the database account that you created for Oracle Identity Manager.
   - In the **Password** field, enter the Oracle Identity Manager database user password.

- Click **Next** to commit these settings.

  Note: When you set the preceding items, see the configuration settings specified in "Using an Oracle Database for Oracle Identity Manager" on page 4-1 to verify your settings.

  The installer checks for database connectivity and if a database schema exists. If the check passes, the installer proceeds to the next step in the process. If the check fails, an error message is displayed.

- Select the appropriate database options:
  - If a database exists, and the connectivity is good, proceed to Step 10.
  - If no connectivity is detected, you are prompted to enter new information or to fix the connection. Click **Next** after entering new information or fixing the connection.

10. On the Authentication Information page, select either the **Oracle Identity Manager Default Authentication** or **SSO** (Single Sign-On) **Authentication** option. If you select Single Sign-On authentication, you must provide the header variable used in the Single Sign-On system in the **Enter the header value for SSO Authentication** field. Click **Next**.
11. On the Application Server Selection page, select **Oracle Application Server**, then click **Next**.
12. Specify the server configuration by selecting **No** on the Application Server is Clustered page and click **Next**. Refer to Chapter 9, "Deploying in a Clustered Oracle Application Server Configuration" if you are deploying Oracle Identity Manager for an Oracle Application Server cluster.
13. On the Application Server Information page, enter the information about your application server and Java installation:
    a. Enter the path to your application server installation directory.

       Alternatively, click **Browse** and navigate to your application server installation directory.

    b. Enter the path to the Oracle Application Server JDK directory.

       Alternatively, click **Browse** and navigate to the Oracle Application Server JDK directory.

    c. Click **Next**.

    d. Enter the Oracle Application Server administrator user name, oc4jadmin, in the **User Name** field.

    e. Enter the password for the OC4J administrator in the **Password** field.

    f. Enter the OC4J instance name in the **OC4J Instance Name** field.

    g. Enter the RMI port number in the **RMI Port No** field. You can identify the RMI port number by executing the following command from the *ORACLE_HOME*\opmn\bin\ directory:

       ```
       opmnctl status -l
       ```

14. If you have not backed up the application server, then create a backup of the application server when the Application Server Configuration Backup page is displayed. Then click **Next** to start server installation.

15. If the installer detects an existing database, you can choose to use that database. Select **Yes**, then click **Next**. If the existing database is not encrypted, you are prompted to encrypt it. Select **Yes**, then click **Next**.
16. The Summary page is displayed. Click **Install** to install the application.
17. The Completed page is displayed. Click **Finish** to exit the installer.

After installing Oracle Identity Manager, follow the instructions in Chapter 7, "Postinstallation Configuration for Oracle Identity Manager and Oracle Application Server".

Removing Oracle Identity Manager

To remove an Oracle Identity Manager installation:

1. Stop Oracle Identity Manager if it is running and stop all Oracle Identity Manager processes by stopping Oracle Application Server.
2. Delete the *OIM_HOME* directory in which you installed Oracle Identity Manager.

6

Installing Oracle Identity Manager Server on UNIX

This chapter explains how to install Oracle Identity Manager on UNIX in a nonclustered installation.

See Also:

- *Oracle Identity Manager Release Notes* for information about supported UNIX platforms
- Chapter 9, "Deploying in a Clustered Oracle Application Server Configuration" for information about deploying Oracle Identity Manager in a clustered installation

You must install Oracle Identity Manager on systems running Oracle Application Server. Oracle Identity Manager components such as the Remote Manager can be installed on separate systems. Each component has its own installer.

This chapter contains the following topics:

Installation Prerequisites and Notes

The following is a list of prerequisites and notes for installing Oracle Identity Manager on UNIX:

- The Oracle Identity Manager Installer program requires at least 200 MB of free space in the home directory while installing Oracle Identity Manager. Check the /etc/passwd file to determine the home directory. Note that you cannot work around this requirement by changing the value of the $HOME variable.
- There must be at least 200 MB of free space in the /var/tmp directory.
- Before installing Oracle Identity Manager, you must set the JAVA_HOME variable to point to Oracle Application Server JDK. For details, go to "Setting Environment Variables" on page 3-4 and see the section "For UNIX".

- The default logging package included by the base RedHat Linux installation causes installation problems and exceptions for Oracle Identity Manager. Before installing Oracle Identity Manager on RedHat Linux, delete the commons-logging-1.0.2 library from the base operating system installation. The commons-logging-1.0.2 library is typically installed with any standard RedHat installation. Also, ensure that you delete the symbolic links in the `/usr/share/java/` directory. Deleting these symbolic links will force Oracle Identity Manager to use its own internal logger JAR files during installation.
- Set the Java binary in the PATH variable for Oracle Identity Manager Installer to work effectively. For details, refer to "Setting Environment Variables" on page 3-4 and see the section "For UNIX".
- During the installation process, an unused log file named log.conf is created in the *`OIM_HOME`*`/xellerate/config/` directory.
- Do not install Oracle Identity Manager on top of an existing Oracle Identity Manager installation. Use a different Oracle Identity Manager home directory. If you want to reuse the same directory name for the Oracle Identity Manager home directory then back up your previous Oracle Identity Manager home by renaming the original directory.

  In addition, all Oracle Identity Manager components must be installed in different home directories. For example, you cannot install the Remote Manager in the same directory in which Oracle Identity Manager is installed.

Installing the Database Schema

As part of the installation, the Oracle Identity Manager Installer loads a schema into the database. You only install the database schema once. It is installed the first time you run the Oracle Identity Manager Installer. Each subsequent time you run the installer to deploy other Oracle Identity Manager components, you enter information about the database connection to configure the component for the same schema. If required, contact your database administrator (DBA).

Note: During the schema installation, a log file is created in the *`OIM_HOME`*`/logs` directory.

Installing Documentation

The Oracle Identity Manager documentation is installed automatically in the *`OIM_HOME`* directory. No special input is required. A full documentation set is installed with each Oracle Identity Manager component.

Installing Oracle Identity Manager on UNIX

To install Oracle Identity Manager on the Oracle Application Server running on UNIX, you must install Oracle Identity Manager as the same non-root user who installed the Oracle Application Server. Do not attempt to install Oracle Identity Manager on the Oracle Application Server running on UNIX as the root user.

Oracle Identity Manager for UNIX is installed through a console mode installer, which supports the following two input methods:

- Choose from among list of options

Each option is numbered and accompanied by brackets ([ ]). To select an option, enter its number. Once selected, the associated brackets display an X ([X]).

- Enter information at a prompt

  Type the information at the prompt and press **Enter**.

  Default values are enclosed in brackets after a prompt; to accept a default value, press **Enter**.

The installer contains logical sections (panels). You can perform the following actions in the panels:

- When you have selected an item from a list of options, enter zero (**0**) to indicate that the desired item has been selected.
- To move to the next installation panel, enter **1**.
- To go back to the previous panel, enter **2**.
- To cancel the installation, enter **3**.
- To redisplay the current panel, enter **5**.

To install Oracle Identity Manager on UNIX:

> **Note:** Create a backup of the Oracle Application Server configuration before installing Oracle Identity Manager. For more information, see "Creating a Backup of the Oracle Application Server Configuration" on page 3-5.

1. Insert the Oracle Identity Manager Installation CD into your CD-ROM drive.
2. From the console, change directory (cd) to the installServer directory on the installation CD and run the install_server.sh file by using the following command:

   ```
   sh install_server.sh
   ```

   The installer starts in console mode.

   > **Note:** If you are not installing Oracle Identity Manager from distributed media (CD), you must set the execute bit of all shell scripts in the installServer directory. To set the execute bit for all shell scripts recursively, cd to the installServer directory and run the following command:
   >
   > ```
   > find . -name "*.sh" -exec chmod u+x {} \;
   > ```

3. Choose a language by entering a number from the list of languages.

   Enter **0** to apply the language selection. The Welcome Message panel is displayed.

4. Enter **1** on the Welcome Message panel to display the next panel.

   The Admin User Information panel is displayed

5. Enter a password you want to use for the Oracle Identity Manager Administrator, confirm the password by entering it again, and then enter **1** to move to the next panel.

   The OIM Application Options panel is displayed.

6. Enter **1** on the OIM Application Options panel to display the next panel.

   The Select the Oracle Identity Manager application to install panel is displayed.

7. Select the application to install:

   - Enter **1** for Oracle Identity Manager.
   - Enter **2** for the Oracle Identity Manager with Audit and Compliance Module.

   Enter 0 when you are finished to apply the application selection. The Target directory panel is displayed.

8. On the Target directory panel, enter the path to the directory in which you want to install Oracle Identity Manager. For example, enter `/opt/oracle/`. Enter **1** to move to the next panel.

   Important: Do not install Oracle Identity Manager on top of an existing Oracle Identity Manager installation. Use a different Oracle Identity Manager home directory. If you want to reuse the same directory name for the Oracle Identity Manager home directory backup your previous Oracle Identity Manager home by renaming the original directory.

   All Oracle Identity Manager components must be installed in different home directories. For example, you cannot install the Remote Manager in the same directory where Oracle Identity Manager is installed.

   If the directory does not exist, you are asked to create it. Enter **y** to create the directory.

   The Database Server Selection panel is displayed.

   Note: To install against an existing database, verify that the version of Oracle Identity Manager you are installing is certified with your existing database version. Refer to the *Oracle Identity Manager Release Notes* to confirm the certified configurations.

   When Oracle Identity Manager is installed against an existing database, a warning message is displayed indicating that the database schema already exists and instructing you to copy the .xldatabasekey file from the existing Oracle Identity Manager installation to the new *OIM_HOME*`/xellerate/config` directory after you complete the installation process.

   Create the new *OIM_HOME*`/xellerate/config` directory if it does not already exist.

   ---

9. On the Database Server Selection panel, specify the type of database that you are using:

   - Enter **1** for Oracle Database.
   - Enter **0** when you are finished.
   - Enter **1** to move to the next panel.

Note: Only Oracle database is supported for Oracle Identity manager installation on Oracle Application Server.

10. Enter the database information:
    - Enter the database host name or IP address.
    - Enter the port number, or accept the default.
    - Enter the SID for the database name.
    - Enter the database user name for the account that Oracle Identity Manager uses to connect to the database.
    - Enter the password for the database account that Oracle Identity Manager uses to connect to the database.
    - Enter **1** to move to the next panel.

    The Authentication Information panel is displayed.
11. Select the authentication mode for the Oracle Identity Manager application:
    - Enter 1 for Oracle Identity Manager Default Authentication.
    - Enter 2 for SSO Authentication.
    - Enter 0 when you are finished.

    If you select SSO authentication, you must provide the header variable used in the Single Sign-On system when prompted.

    Enter 1 to move to the next panel.

    The Application Server Selection panel appears.
12. Specify your application server type.
    - Enter **1** for Oracle Application Server.
    - Enter **0** when you are finished.
    - Enter **1** to move to the next panel.

    The Cluster Information panel is displayed.
13. Enter 2 for No (non-clustered). Refer to Chapter 9, "Deploying in a Clustered Oracle Application Server Configuration" if you are deploying Oracle Identity Manager for an Oracle Application Server cluster. Enter 0 to proceed to the next panel.

    The Application Server Information panel appears.
14. In the Application Server Information panel:
    - Enter the path to where the application server is installed
    - Enter the path to where the Oracle Application Server JDK is installed
    - Enter 1 to move to the next section.

    The Oracle Application Server Information panel is displayed.
15. On the Oracle Application Server Information panel:
    - Enter the user name for the Oracle Application Server administrator
    - Enter the password for the Oracle Application Server administrator

- Enter the Oracle Application Server Instance Name
- Enter the RMI port number. You can identify the RMI port number by executing the following command from the *ORACLE_HOME*/opmn/bin/ directory:

  ```
  opmnctl status -l
  ```

16. When you receive a message about backing up the application server installation, enter **1** to move to the next section. The Summary panel is displayed.
17. On the Summary panel, enter **1** to begin installation.
18. After the installation finishes, the Completed panel is displayed. Enter **3** to finish and exit.

After installing Oracle Identity Manager, follow the instructions in Chapter 7, "Postinstallation Configuration for Oracle Identity Manager and Oracle Application Server".

Removing Oracle Identity Manager

To remove an Oracle Identity Manager installation:

1. Stop Oracle Identity Manager if it is running and stop all Oracle Identity Manager processes by stopping Oracle Application Server.
2. Delete the *OIM_HOME* directory in which you installed Oracle Identity Manager.

7

Postinstallation Configuration for Oracle Identity Manager and Oracle Application Server

After installing Oracle Identity Manager, you must complete some postinstallation tasks before using the application. Depending on the deployment, you might choose not to perform some of these tasks.

This chapter discusses the following topics:

- Default JMS Queue Configuration
- Required Postinstallation Tasks
- Optional Postinstallation Tasks

> **Note:** The examples in this chapter are Windows-based, however the postinstallation tasks apply to UNIX as well.

Default JMS Queue Configuration

In releases earlier than 9.1.0, Oracle Identity Manager uses a single JMS queue (named xlQueue) for all asynchronous operations, including, requests, reconciliation, attestation, and offline tasks. Release 9.1.0 onward, by default, Oracle Identity Manager uses separate JMS queues for specific operations to optimize JMS queue processing. The following list shows the JMS queues in the default configuration and indicates the operation related to each queue:

- `xlQueue` (for request operations)
- `xlReconQueue` (for reconciliation operations)
- `xlAuditQueue` (for auditing operations)
- `xlAttestationQueue` (for attestation operations)
- `xlProcessQueue` (for use in a future release)

Required Postinstallation Tasks

After you install Oracle Identity Manager on Oracle Application Server, you must perform the following tasks:

- Changing Keystore Passwords
- Setting the Path of the jgroups-core.jar File in the PurgeCache Script

- Setting Up Database-Based Storage of JMS Queues
- Setting the Compiler Path for Adapter Compilation
- Tuning JDBC Connection Pools
- Increasing the Oracle Application Server Heap Size

Changing Keystore Passwords

During installation, the passwords for the Oracle Identity Manager keystores are set to `xellerate`. The Installer scripts and installation log contain this default password. It is strongly recommended that you change the keystore passwords for all production installations.

To change the keystore passwords, you must change the storepass of .xlkeystore and the keypass of the xell entry in .xlkeystore—and these two values must be identical. Use the keytool and the following steps to change the keystore passwords:

1. Open a command prompt on the Oracle Identity Manager host computer.
2. Navigate to the *OIM_HOME*`\xellerate\config` directory.
3. Run the keytool with the following options to change the storepass:

   ```
   JAVA_HOME\jre\bin\keytool -storepasswd -new new_password -storepass xellerate
   -keystore .xlkeystore -storetype JKS
   ```

4. Run the keytool with the following options to change the keypass of the xell entry in .xlkeystore:

   ```
   JAVA_HOME\jre\bin\keytool -keypasswd -alias xell -keypass xellerate -new
   new_password -keystore .xlkeystore -storepass new_password
   ```

   Note: Replace *new_password* with the same password entered in step 3.

   Table 7–1 lists the options used in the preceding example of keytool usage.

Table 7–1 Command Options for the keytool Utility

Option	Description
JAVA_HOME	Location of the Java directory associated with the application server
new_password	New password for the keystore
`-keystore` *option*	Keystore whose password you are changing (.xlkeystore for Oracle Identity Manager or .xldatabasekey for the database)
`-storetype` *option*	JKS for .xlkeystore and JCEKS for .xldatabasekey

5. Open *OIM_HOME*`\xellerate\config\xlconfig.xml` in a text editor.
6. Edit the <xl-configuration>.<Security>.<XLPKIProvider>.<KeyStore> section, <xl-configuration>.<Security>.<XLPKIProvider>.<Keys> section, and <RMSecurity>.<KeyStore> section to specify the keystore password as follows:

Note: Change the <XLSymmetricProvider>.<KeyStore> section of the configuration file to update the password for the database keystore (.xldatabasekey).

- Change the password tag to `encrypted="false"`.
- Enter the password (in the clear), for example:

```
<Security>
<XLPKIProvider>
<KeyStore>
      <Location>.xlkeystore</Location>
      <Password encrypted="false">new_password</Password>
      <Type>JKS</Type>
      <Provider>sun.security.provider.Sun</Provider>
</KeyStore>
<Keys>
<PrivateKey>
<Alias>xell</Alias>
<Password encrypted="false">new_password</Password>
</PrivateKey>
</Keys>
<RMSecurity>
<KeyStore>
<Location>.xlkeystore</Location>
<Password encrypted="false">new_password</Password>
<Type>JKS</Type>
<Provider>sun.security.provider.Sun</Provider>
</KeyStore>
```

7. Save and close the xlconfig.xml file.
8. Restart the application server.

   When you stop and start the application server, a backup of the configuration file is created. The configuration file (with the new password) is read in, and the password is encrypted in the file.
9. If all of the preceding steps have succeeded, you can delete the backup file.

Note: On UNIX, you might also want to clear the shell's command history by using the following command:

```
history -c
```

Setting the Path of the jgroups-core.jar File in the PurgeCache Script

To set the path of the `jgroups-core.jar` file in the `PurgeCache` script:

See Also: *Oracle Identity Manager Globalization Guide* for information about the `PurgeCache` script

1. Search for the `jgroups-core.jar` file in the Oracle Application Server installation directory.
2. Open the `PurgeCache` file in a text editor.

   For UNIX:

OIM_HOME/xellerate/bin/PurgeCache.sh

For Microsoft Windows:

OIM_HOME\xellerate\bin\PurgeCache.bat

3. Search for the CLASSPATH variable in the PurgeCache file.
4. In the value assigned to the CLASSPATH variable, add the full path and name of the jgroups-core.jar file before %XEL_EXT%\javagroups-all.jar.

Setting Up Database-Based Storage of JMS Queues

The Oracle Identity Manager Installer creates JMS queues for file-based storage of JMS messages. This is the default storage mechanism for JMS queues in Oracle Identity Manager. However, for production environments and clustered installations, it is strongly recommended that you set up database-based storage of JMS queues by performing the procedure described in this section.

Note:

- Refer to Chapter 4, "Using Oracle Enterprise Messaging Service" in Oracle Containers for J2EE Services Guide for information about working with Oracle Enterprise Messaging Service JMS using Advanced Queuing (AQ).
- Create a backup of the Oracle Application Server configuration before setting up database-based storage of JMS queues. For more information, see "Creating a Backup of the Oracle Application Server Configuration" on page 3-5.
- Oracle recommends you to stop all the scheduled tasks by using the Administrative and User Console. Otherwise, you may see AuthenticationException in the server logs in between implementating the AQ instructions and restarting the servers. There is no impact of this exception to the functioning of Oracle Identity Manager, and this exception can be ignored.
- If you are using the AIX operating system, then you must manually undeploy the Xellerate application by using Oracle Application Server Administrative Console before you run the patch_oc4j.cmd or patch_oc4j.sh script.

To set up database-based storage of JMS queues, refer to Note 554624.1 in the following URL:

http://metalink.oracle.com

Setting the Compiler Path for Adapter Compilation

To compile adapters or import Deployment Manager XML files that have adapters, you must set the compiler path. To set the compiler path for adapter compilation, you must first install the Design Console. Refer to Chapter 10, "Installing and Configuring the Oracle Identity Manager Design Console" for instructions on installing the Design Console and then setting the compiler path for adapter compilation.

Tuning JDBC Connection Pools

To implement tuning for the JDBC connection pools used by Oracle Identity Manager, open *ORACLE_HOME*/j2ee/*INSTANCE_NAME*/config/data-sources.xml file and implement the following changes:

> **Note:** It is strongly recommended that you implement the suggested tuning for the JDBC connection pools used by Oracle Identity Manager. This can be further tuned based on the application usage.

1. For xlConnectionPool, the minimum and maximum connection pool values should be set as follows:

   ```
   min-connections="10"
   max-connections="50"
   ```

2. For `xlXAConnectionPool`, the minimum and maximum connection pool values should be set as follows:

   ```
   min-connections="30"
   max-connections="100"
   ```

3. After implementing the changes, restart the Oracle Application Server instance for the change to take effect.

> **Note:** For clustered installation of Oracle Identity Manager on Oracle Application Server, the changes mentioned in steps 1 through 3 can be implemented for all the Oracle Application Server instances. Also make sure that the database supports the increase in the number of connections.

Increasing the Oracle Application Server Heap Size

After installing Oracle Identity Manager on Oracle Application Server, you must change the JVM memory settings for production environments or when you are processing large volume in non-production.

Perform the following steps to increase the Oracle Application Server heap size:

1. Open the *ORACLE_HOME*\opmn\conf\opmn.xml file in a text editor.
2. Change the memory setting for the OC4J instance where Oracle Identity Manager is installed from:

   ```
   -ms512M -mx1024M
   ```

   To:

   ```
   -ms1280m -mx1280m
   ```

3. Save and close the *ORACLE_HOME*\opmn\conf\opmn.xml file.
4. Restart the Oracle Application Server.

Optional Postinstallation Tasks

After installing Oracle Identity Manager, consider performing the following optional postinstallation tasks documented in this section before using the application.

Depending on the Oracle Identity Manager deployment, you may choose not to perform some of these tasks.

- Setting Log Levels
- Enabling Single Sign-On (SSO) for Oracle Identity Manager
- Deploying the SPML Web Service
- Changing Transaction Timeout

Setting Log Levels

Oracle Identity Manager uses log4j for logging. Logging levels are configured in the logging properties file, *OIM_HOME*/xellerate/config/log.properties. By default, the log level is set to Warning, except for DDM, for which the log level is set to Debug by default. You can change the log level universally for all components or for an individual component.

Oracle Identity Manager components are listed in the *OIM_HOME*\xellerate\config\log.properties file in the XELLERATE section, for example:

```
log4j.logger.XELLERATE=WARN
log4j.logger.XELLERATE.DDM=DEBUG
log4j.logger.XELLERATE.ACCOUNTMANAGEMENT=DEBUG
log4j.logger.XELLERATE.SERVER=DEBUG
log4j.logger.XELLERATE.RESOURCEMANAGEMENT=DEBUG
log4j.logger.XELLERATE.REQUESTS=DEBUG
log4j.logger.XELLERATE.WORKFLOW=DEBUG
log4j.logger.XELLERATE.WEBAPP=DEBUG
log4j.logger.XELLERATE.SCHEDULER=DEBUG
log4j.logger.XELLERATE.SCHEDULER.Task=DEBUG
log4j.logger.XELLERATE.ADAPTERS=DEBUG
log4j.logger.XELLERATE.JAVACLIENT=DEBUG
log4j.logger.XELLERATE.POLICIES=DEBUG
log4j.logger.XELLERATE.RULES=DEBUG
log4j.logger.XELLERATE.DATABASE=DEBUG
log4j.logger.XELLERATE.APIS=DEBUG
log4j.logger.XELLERATE.OBJECTMANAGEMENT=DEBUG
log4j.logger.XELLERATE.JMS=DEBUG
log4j.logger.XELLERATE.REMOTEMANAGER=DEBUG
log4j.logger.XELLERATE.CACHEMANAGEMENT=DEBUG
log4j.logger.XELLERATE.ATTESTATION=DEBUG
log4j.logger.XELLERATE.AUDITOR=DEBUG
```

To set Oracle Identity Manager log levels, edit the logging properties in the *OIM_HOME*\xellerate\config\log.properties file as follows:

1. Open the *OIM_HOME*\xellerate\config\log.properties file in a text editor. This file contains a general setting for Oracle Identity Manager and specific settings for the components and modules that comprise Oracle Identity Manager.

   By default, the log level in Oracle Identity Manager is set to Warning:

   ```
   log4j.logger.XELLERATE=WARN
   ```

   This is the general value for Oracle Identity Manager. Individual components and modules are listed following the general value in the properties file. You can set individual components and modules to different log levels. The log level for a specific component overrides the general setting.

2. Set the general value to the desired log level. The following is a list of the supported log levels, appearing in descending order of information logged (DEBUG logs the most information and FATAL logs the least information):
   - DEBUG
   - INFO
   - WARN
   - ERROR
   - FATAL
3. Set other component log levels as desired. Individual components or modules can have different log levels. For example, the following values set the log level for the Account Management module to INFO, while the server is at DEBUG and the rest of Oracle Identity Manager is at the WARN level.

   ```
   log4j.logger.XELLERATE=WARN

   log4j.logger.XELLERATE.ACCOUNTMANAGEMENT=INFO

   log4j.logger.XELLERATE.SERVER=DEBUG
   ```
4. Save your changes.
5. Restart your application server so that the changes take effect.

Enabling Single Sign-On (SSO) for Oracle Identity Manager

The following procedure describes how to enable Single Sign-On for Oracle Identity Manager with ASCII character logins. To enable Single Sign-On with non-ASCII character logins, use the following procedure—but include the additional configuration setting described in Step 4.

> **See Also:** *Oracle Identity Manager Best Practices Guide* for additional information about configuring Single Sign-On for Oracle Identity Manager with Oracle Access Manager.

> **Note:** Header names comprised only of alphabetic characters are certified. Oracle recommends that for not using special characters or numeric characters in header names.

To enable Single Sign-On for Oracle Identity Manager:

1. Stop the application server gracefully.
2. Open *OIM_HOME*/xellerate/config/xlconfig.xml in a text editor.
3. Locate the following Single Sign-On configuration (the following are the default settings without Single Sign-On):

   ```
   <web-client>
   <Authentication>Default</Authentication>
   <AuthHeader>REMOTE_USER</AuthHeader>
   </web-client>
   ```
4. Edit the Single Sign-On configuration to the following and replace *SSO_HEADER_NAME* with the appropriate header configured in your Single Sign-On system:

```
<web-client>
<Authentication>SSO</Authentication>
<AuthHeader>SSO_HEADER_NAME</AuthHeader>
</web-client>
```

To enable Single Sign-On with non-ASCII character logins, you must include a decoding class name to decode the non-ASCII header value. Add the decoding class name and edit the Single Sign-On configuration as follows:

```
<web-client>
<Authentication>SSO</Authentication>
<AuthHeader>SSO_HEADER_NAME</AuthHeader>
<AuthHeaderDecoder>com.thortech.xl.security.auth.CoreIDSSOAuthHeaderDecoder</Au
thHeaderDecoder>
</web-client>
```

Replace *SSO_HEADER_NAME* with the appropriate header configured in your Single Sign-On system.

5. Change your application server and Web server configuration to enable Single Sign-On by referring to your application and Web server vendor documentation.
6. Restart the application server.

Deploying the SPML Web Service

Organizations can have multiple provisioning systems that exchange information about the modification of user records. In addition, there can be applications that interact with multiple provisioning systems. The SPML Web Service provides a layer over Oracle Identity Manager to interpret SPML requests and convert them to Oracle Identity Manager calls.

The SPML Web Service is packaged in a deployable Enterprise Archive (EAR) file. This file is generated when you install Oracle Identity Manager.

Because the EAR file is generated while you install Oracle Identity Manager, a separate batch file in the Oracle Identity Manager home directory runs the scripts that deploy the SPML Web Service on the application server on which Oracle Identity Manager is running. You must run the batch file to deploy the SPML Web Service.

For details about the SPML Web Service, see Chapter 12, "The SPML Web Service" in *Oracle Identity Manager Tools Reference*.

Changing Transaction Timeout

The default value of import and export operations transaction timeout is 600 seconds. This timeout overrides the default global transaction timeout of 1200 seconds.

To increase or modify transaction timeout for export and import operations:

1. Go to the *OIM_HOME*`/xellerate/DDTemplates/DO` directory.
2. Open the `orion-ejb-jar.xml` file.
3. In the `orion-ejb-jar.xml` file, search for **transaction-timeout**. You will find transaction-timeout in two places in the file, one for import and another for export.
4. Increase or modify the value of the transaction-timeout.
5. Go to the *OIM_HOME*`/xellerate/setup` directory.
6. Based on the operating system on which Oracle Identity Manager is installed, run `patch_oc4j.sh` or `patch_oc4j.cmd`.

Note:

- For a clustered installation of Oracle Identity Manager, the steps to increase or modify transaction timeout for export and import operations must be repeated for all the Oracle Application Server instances.
- If HTTP server timeout is lower than the value you set for Oracle Identity Manager, then you might need to increase the HTTP server timeout value. The timeout value of HTTP Server must be higher then the timeout value of Oracle Identity Manager.

8

Starting and Stopping Oracle Identity Manager

This chapter explains how to start and stop Oracle Identity Manager, and how to access the Administrative and User Console. This chapter contains the following topics:

> **Important:** You must complete all post-installation steps in Chapter 7, "Postinstallation Configuration for Oracle Identity Manager and Oracle Application Server" on page 7-1 before starting Oracle Identity Manager.

Removing Backup xlconfig.xml Files After Starting or Restarting

After you start any Oracle Identity Manager component for the first time, or after you change any passwords in the xlconfig.xml file, Oracle Identity Manager encrypts and saves the passwords. Oracle Identity Manager also creates a backup copy of the xlconfig.xml file before saving changes to the file. These backup files contain old passwords in plaintext. The backup file are named xlconfig.xml.*x*, where *x* is the latest available number, for example xlconfig.xml.0, xlconfig.xml.1, and so on.

> **Note:** You must remove these backup files after starting any Oracle Identity Manager component for the first time, or on restarting after changing any passwords in xlconfig.xml once you have established that the new password is working properly.

Starting Oracle Identity Manager

This section describes how to start Oracle Identity Manager on Microsoft Windows, UNIX. To start Oracle Identity Manager, start Oracle Application Server. Use the following steps to start the Oracle Application Server and Oracle Identity Manager:

1. Verify that your database is up and running

2. Start Oracle Application Server as follows:

   Microsoft Windows

   From the **Start** menu, select **Oracle -** ***Oracle Application Server Instance Name***, Oracle **Process Manager**, and then click **Start Oracle Process Manager**.

   UNIX

   a. Go to the *ORACLE_HOME*/opmn/bin directory.

   b. Run the following command:

   ```
   ./opmnctl startall
   ```

Stopping Oracle Identity Manager

To stop Oracle Identity Manager gracefully, stop the Oracle Application Server by performing the following steps:

Microsoft Windows

From the **Start** menu, select **Oracle -** ***Oracle Application Server Instance Name***, then select Oracle **Process Manager**, then select **Stop Oracle Process Manager**.

UNIX or Linux

1. Go to the *ORACLE_HOME*/opmn/bin/ directory.
2. Run the following command:

   ```
   ./opmnctl stopall
   ```

Accessing the Administrative and User Console

After starting Oracle Application Server and Oracle Identity Manager, you can access the Administrative and User Console. Perform the following steps to access the Administrative and User Console:

1. Add the fully-qualified domain name of the host with your Oracle Application Server instance to your system's hosts file. This fully-qualified domain name can be found from URL that appears after you log into Enterprise Manager for Oracle Application Server.
2. Browse to the following URL by using a Web browser:

   ```
   http://hostname:port/xlWebApp
   ```

   In this URL, *hostname* represents the name of the computer hosting the application server and *port* refers to the port on which the server is listening. The default port number for Oracle Application Server is 7777. To determine the port that Oracle Application Server is listening on, open *ORACLE_HOME*/install/readme.txt on UNIX or Linux, and *ORACLE_HOME*\install\readme.txt on Microsoft Windows.

   Note: The application name, xlWebApp, is case-sensitive.

   For example:

   ```
   http://localhost:7777/xlWebApp
   ```

3. After the Oracle Identity Manager login page is displayed, log in with your user name and password.

Using the Diagnostic Dashboard to Verify Installation

The Diagnostic Dashboard verifies each component in your postinstallation environment by testing for:

- A trusted store
- Single Sign-On Configuration
- Messaging capability
- A task scheduler
- A Remote Manager

The Diagnostic Dashboard also checks for all supported versions of components along with their packaging.

Note: See "Using the Diagnostic Dashboard" on page 2-4 for more information.

9

Deploying in a Clustered Oracle Application Server Configuration

This chapter describes how to deploy Oracle Identity Manager in a clustered Oracle Application Server environment and contains the following sections:

Overview of Deploying in a Clustered Oracle Application Server Configuration

This section describes the steps to deploy Oracle Identity Manager in a Oracle Application Server clustered configuration by using three example cluster members as follows:

- Cluster Node A and Cluster Node B are Oracle Application Servers running Oracle Identity Manager.
- Cluster Node C is an Oracle HTTP Web Server.

Installing Oracle Application Server on Cluster Members

The first step in deploying Oracle Identity Manager in a clustered Oracle Application Server configuration is to install Oracle Application Server on the cluster members, for example, on Cluster Node A, Cluster Node B, and Cluster Node C.

Perform the following steps on the Oracle Universal Installer to install Oracle Application Server on the cluster members. When you run the Oracle SOA Suite installer, you must choose the Advanced Install option.

- On Cluster Node A:
  - Install Oracle Application Server by selecting the **J2EE Server** option on the Select Installation Type page.
  - In the Specify Port Configuration Options window that is displayed, **Automatic** is selected by default. Click **Next**.
  - Select the **Configure this as an Administrator OC4J instance** option on the Administration Settings page.
  - Select the **Configure this OC4J instance to be part of an Oracle Application Server cluster topology** option and specify the host IP address and Port for the discovery address on the Cluster Topology Configuration page.

    Note:

    - Discovery address (Multicast address and port number) for Cluster Node A, Cluster Node B, and Cluster Node C must be the same.

    - The path of *ORACLE_HOME* has to be same for Node A and Node B.

- On Cluster Node B:
  - Install Oracle Application Server by selecting the **J2EE Server** option on the Select Installation Type page.
  - In the Specify Port Configuration Options window that is displayed, **Automatic** is selected by default. Click **Next**.
  - ***Do not*** select the **Configure this as an Administrator OC4J instance** option on the Administration Settings page.
  - Select the **Configure this OC4J instance to be part of an Oracle Application Server cluster topology** option and specify the host IP address and Port for the discovery address on the Cluster Topology Configuration page.
- On Cluster Node C:
  - Install the Web server by selecting the **Web Server** option on the Select Installation Type page.
  - In the Specify Port Configuration Options window that is displayed, **Automatic** is selected by default. Click **Next**.
  - Select the **Configure this Oracle HTTP Server instance to be part of an Oracle Application Server cluster** option and specify the host IP address and Port for the discovery address on the Cluster Topology Configuration page.

Upgrading Oracle Application Server From Release 10.1.3.1 To Release 10.1.3.3

Ensure that you have upgraded Oracle Application Server from release 10.1.3.1 to release 10.1.3.3. To check the version of Oracle Application Server that you currently use:

1. Ensure that the Oracle Application Server Java binary path is *ORACLE_HOME*\jdk\bin.
2. Run the following command from *ORACLE_HOME*\j2ee*OC4J_INSTANCE*\java -jar oc4j.jar version.

   The information that appears should be similar to the following example:

   ```
   Oracle Containers for J2EE 10g (10.1.3.3.0) (build
   070610.1800.23513)
   ```

Applying Oracle Application Server Patches

The following patches may be specific to 10.1.3.3.0 version of Oracle Application Server:

Note:

- If you are running on any other version of Oracle Application Server, then contact Oracle to get the patches for that version of Oracle Application Server.
- If you are installing Oracle Identity Manager in Microsoft Windows Vista, then you must set the environment variable OPATCH_PLATFORM_ID to 207.

1. Install Oracle Application Server patch 2617419.
2. Install Oracle Application Server patch 6685235.
3. Install Oracle Application Server patch 5389650.
4. Install Oracle Application Server patch 6454278.

   Patches and instructions on how to apply those patches can be downloaded from the Oracle*MetaLink* Web site at:

   http://metalink.oracle.com.

Note: All the steps to install Oracle Application Server patches must be repeated for all installations of Oracle Application Server.

Creating OC4J Instances

After installing Oracle Application Server on the cluster members, create OC4J instances named xlClusterMember1 on Cluster Node A. For this:

1. Log in to the Oracle Enterprise Manager 10g Application Server Control.
2. Click the name of the Cluster Node you want to create the OC4J instance on.
3. Click **Create OC4J Instance**.
4. Enter xlClusterMember1 as the instance name and select **Add to a new group with name** and enter xlClusterGroup as the Group Name.
5. Click Create to create and start the xlClusterMember instance.
6. Start the OC4J instances you created on each Cluster Node.

Repeat these steps for creating `xlClusterMember2` on Cluster Node B by using appropriate values. Use `xlClusterGroup` as the Group Name.

Max Server Sockets for OC4J Instance

To avoid `OutOfMemory` errors during runtime, open *`ORACLE_HOME`*`/j2ee/`*`OC4J_INSTANCE`*`/config/rmi.xml`, and add `max-server-sockets="200"` as shown:

```
<rmi-server
...
max-server-sockets="200"
>
```

Note:

- This configuration is part of patch 6685235.
- *`OC4J_INSTANCE`* is the instance of Oracle Application Server where Oracle Identity Manager is deployed. For a clustered installation of Oracle Identity Manager, this needs to be repeated for `xlClusterMember1` and `xlClusterMember2`.

Setting the RMI Port Number Range

The Oracle Process Manager and Notification server (OPMN) dynamically assigns port numbers to each OC4J instance within your Oracle Application Server instance.

To ensure that you can access the Oracle Identity Manager Design Console and Administrative and User Console on Oracle Application Server, you must set the RMI port number range to be unique in the *`ORACLE_HOME`*`/opmn/conf/opmn.xml` file. Perform the following steps to set a unique RMI port number range in each node:

1. In a text editor, open the *`ORACLE_HOME`*`/opmn/conf/opmn.xml` file.
2. Locate the `<port id="rmi" range="12401-12500"/>` entry for cluster member xlClusterMember1 in Node A.
3. Choose one of the port for RMI within the range of 12401 and 12500, for example, `<port id="rmi" range="12409"/>`. Make sure that you use this same RMI port ID for the clustergroup in other nodes.
4. Save and close the opmn.xml file.

Note: Repeat the steps 1 through 4 for the all the other cluster members, such as xlClusterMember2 in Node B.

5. Stop and restart the Oracle Application Servers for the changes to take effect. Verify the change by running the `opmnctl status -l` command from the *`ORACLE_HOME`*`/opmn/bin/` directory.

Note: Refer to the "Setting Environment Variables" section on page 3-4 for information about setting environment variables.

Installing and Configuring a Database for Oracle Identity Manager

Refer to Chapter 4, "Installing and Configuring a Database for Oracle Identity Manager" for information.

Installing Oracle Identity Manager on Cluster Members

After creating OC4J instances, install Oracle Identity Manager on both Cluster Node A and then Cluster Node B with the same database schema information.

Refer to Chapter 5, "Installing Oracle Identity Manager on Microsoft Windows" or Chapter 6, "Installing Oracle Identity Manager Server on UNIX" to install Oracle Identity Manager on both Cluster Node A and Cluster Node B.

Important: When installing Oracle Identity Manager on both Cluster Node A and Cluster Node B, ensure that you set the following information when prompted by the Oracle Identity Manager Installer:

- Select **Yes** when prompted as to whether the application server is clustered.
- Enter **xlClusterGroup** as Cluster Name.
- Enter the appropriate OC4J instance name (`xlClusterMember1` or `xlClusterMember2`).
- Enter xlClusterMember's RMI port number as the RMI port. You can identify the RMI port number by executing the following command from the *`ORACLE_HOME`*`/opmn/bin/` directory:

  ```
  opmnctl status -l
  ```

Configuring Oracle Identity Manager for the Oracle Application Server Cluster

After installing Oracle Identity Manager, perform the following steps to configure it for the Oracle Application Server cluster:

1. Copy the Oracle Identity Manager log4j-1.2.8.jar log file from the *`OIM_HOME`*`/xellerate/ext/` directory to the *`ORACLE_HOME`*`/jdk/jre/lib/ext/` directory on both Cluster Node A and Cluster Node B.
2. Restart all the cluster members, including Cluster Node A, Cluster Node B, and Cluster Node C.

Accessing the Oracle Identity Manager Administrative and User Console for the Cluster

After restarting all the cluster members, you can access the Oracle Identity Manager Administrative and User Console for the cluster by going to the following URL:

`http://`*`Node_C_host_name`*`:`*`web_server_port`*`/xlWebApp`

Note: *Node_C_host_name* represents the host name of the Web server and *web_server_port* represents the port number of the server on Node C, which is 7777 by default.

Installing and Configuring the Design Console for the Cluster

You can install the Oracle Identity Manager Design Console on any Windows node in the cluster. Refer to Chapter 10, "Installing and Configuring the Oracle Identity Manager Design Console" for Design Console requirements and installation steps.

Configuring the Design Console to be Cluster Aware

You must configure the Design Console to be cluster aware. Perform the following steps to configure the Design Console to be cluster aware:

1. Open *OIM_DC_HOME*/xlclient/config/xlconfig.xml in a text editor.
2. Modify the java.naming.provider.url attribute to be cluster aware, for example:

   ormi://*host name*:12408/Xellerate,ormi://*host name*:12408/Xellerate

3. Save and close the xlconfig.xml file.

Postinstallation Tasks for Clustered Oracle Application Server Installation

Before the postinstallation configuration for a clustered installation of Oracle Application Server, follow the instructions in Chapter 7, "Postinstallation Configuration for Oracle Identity Manager and Oracle Application Server".

Postinstallation configuration for clustered Oracle Application Server installation involves synchronizing the multicast IP for all Oracle Identity Manager installations. To do this:

1. Go to *ORACLE_HOME*/j2ee/*OC4J_INSTANCE*/application-deployments/Xellerate/ in the first node.
2. Open orion-application.xml.
3. Search for multicast IP and record the IP address.
4. Go to *ORACLE_HOME*/j2ee/*OC4J_INSTANCE*/application-deployments/Xellerate/ in the second node.
5. Open orion-application.xml.
6. Search for multicast IP and change it to the same multicast IP value as the first node.
7. Apply step 6 to all other nodes.
8. Go to *OIM_HOME*/xellerate/config/ in the second node.
9. Open xlconfig.xml and search for *MultiCastAddress* and change it to the same multicast IP value as the first node.

Note: There are two instances of `MultiCastAddress` in `xlconfig.xml` that are used by Oracle Identity Manager for internal caching and scheduler-related activities. These values must be the same. Also, cluster members `xlClusterMember1` and `xlClusterMember2` must have the same value for `MultiCastAddress` in `xlconfig.xml`.

---

10. Apply step 9 to all other nodes.
11. Restart all servers.

Note: You must synchronize multicast IP address in orion-application.xml every time after you run the `patch_oc4j` script. Multicast address in orion-application.xml is used by Oracle Application Server for HTTP session replication and failover.

10

Installing and Configuring the Oracle Identity Manager Design Console

This section explains how to install the Oracle Identity Manager Design Console Java client. You have the option to install the Design Console on the same computer as your Oracle Identity Manager server or on a separate computer.

This chapter contains the following topics:

Requirements for Installing the Design Console

Verify that your environment meets the following requirements for Design Console installation:

- You must have an Oracle Identity Manager server installed and running.
- If you are installing the Design Console on a computer other than the host for the application server, you must know the host name and port number of the computer hosting that application server.
- The Design Console host must be able to ping the application server host by using both IP and host name.

> **Note:** If you cannot resolve the host name of the application server, then try adding the host name and IP address in the hosts file in the directory C:\winnt\system32\drivers\etc\.

Installing the Design Console

To install the Design Console on a Microsoft Windows host:

1. Insert the Oracle Identity Manager Installation CD into your CD-ROM drive.

2. Using Windows Explorer, navigate to the installServer directory on the installation CD.
3. Double-click the setup_client.exe file.
4. Choose a language from the list on the Installer page. The Welcome page is displayed.
5. On the Welcome page, click **Next**.
6. On the Target directory page, complete one of the following sub-steps:

   Note: All Oracle Identity Manager components must be installed in different home directories. If you are installing the Design Console on a computer that is hosting another Oracle Identity Manager component, such as Oracle Identity Manager or the Remote Manager, you must specify a different installation directory for the Design Console.

   a. The default directory for the Design Console is C:\oracle. To install the Design Console in this directory, click **Next**.
   b. To install the Design Console in another directory, specify the path of the directory in the **Directory** field, and then click **Next**.

   Note: If the directory path that you select does not exist, then the Base Directory settings field is displayed. Click **OK**. The directory is automatically created. If you do not have write permission to create the default directory for Oracle Identity Manager, then a message is displayed informing you that the installer could not create the directory. Click **OK** to close the message and then contact your system administrator to obtain the appropriate permissions.

7. On the Application Server page, select **Oracle Application Server**, and then click **Next**. The next page prompts you to specify the JRE to use with Design Console.
8. Select the JRE that is installed with Oracle Identity Manager or specify an existing JRE. Then, click **Next**. The Application Server configuration page is displayed.
9. On the Application Server Host Information page, enter the information appropriate for the application server hosting your Oracle Identity Manager server:
   a. Enter the host name or IP address in the upper field.
   b. Use the default value of 12401 for the Oracle Application Server naming port or specify the appropriate value you set for the Oracle Application Server.

   Note: The host name is case-sensitive.

   c. Click **Next**.
10. On the Graphical Workflow Rendering Information page, enter the Application server configuration information:
    a. Enter the Oracle Identity Manager server (host) IP address.

b. Enter the port number at which Oracle HTTP Server is listening, the default port being 7777. To determine the port that Oracle HTTP Server is listening on, open *ORACLE_HOME*/install/readme.txt on UNIX or Linux, and *ORACLE_HOME*\install\readme.txt on Microsoft Windows.

c. Select **No** to specify whether or not the Design Console must use Secure Sockets Layer (SSL).

d. Click **Next**.

11. On the Shortcut page, select (or deselect) the check boxes for the shortcut options according to your preferences:

a. Choose to create a shortcut to the Design Console on the Start Menu.

b. Choose to create a shortcut to the Design Console on the desktop.

c. Click **Next** when you are satisfied with the check box settings.

12. On the Summary page, click **Install** to initiate Design Console installation.

13. The final installation page displays a reminder to copy certain application server-specific files to your Oracle Identity Manager server installation. Follow these instructions and then click OK.

14. Click **Finish** to complete the installation process.

Postinstallation Requirements for the Design Console

After installing the Design Console, you must perform the following steps before using it for Oracle Identity Manager on Oracle Application Server:

1. Copy the *ORACLE_HOME*\j2ee\home\lib\ejb.jar file on the Oracle Application Server system to the *OIM_DC_HOME*\xlclient\ext directory on the Design Console system.

2. Copy the *ORACLE_HOME*\j2ee\home\oc4jclient.jar file on the Oracle Application Server system to the *OIM_DC_HOME*\xlclient\ext directory on the Design Console system.

3. In the configuration XML file, change the multicast address to match that of Oracle Identity Manager:

a. Open the following file:

OIM_HOME\xellerate\config\xlconfig.xml

b. Search for the <MultiCastAddress> element, and copy the value assigned to this element.

c. Open the following file:

OIM_DC_HOME\xlclient\Config\xlconfig.xml

d. Search for the <Cache> element, and replace the value of the <MultiCastAddress> element inside this element with the value that you copy in Step b.

Starting the Design Console

To start the Design Console, double-click *OIM_DC_HOME*\xlclient\xlclient.cmd or select Design Console from the Windows Start menu or desktop to start the Design Console.

Setting the Compiler Path for Adapter Compilation

In the System Configuration form of the Design Console, you must set the `XL.CompilerPath` system property to include the path of the bin directory inside the JDK directory (*`JDK_HOME`*`\bin`) that is used by the application server on which Oracle Identity Manager is deployed.

Then, restart Oracle Identity Manager.

> **See Also:** The "Rule Elements, Variables, Data Types, and System Properties" section in *Oracle Identity Manager Reference*

Enabling SSL Communication (Optional)

After installing the Oracle Identity Manager Design Console, you might want to configure it to communicate with Oracle Identity Manager server by using SSL. Use the following procedure to complete this task. This involves a two step process in which the communication channel to Oracle HTTP Server and Oracle Application Server instances are secured.

- Enabling SSL for HTTP Communication to Oracle HTTP Server
- Enabling SSL for RMI Communication to Oracle Application Server Instances

The following sections provide information required for enabling SSL communication between the Design Console and Oracle Application Server.

Prerequisites or Assumptions

The following are the prerequisites or assumptions for enabling SSL communication:

- The default certificate store *`ORACLE_HOME`*`\Apache\Apache\conf\ssl.wlt\default\ewallet.p12` is being used by the Oracle HTTP Server. The password for the store must be `welcome`.
- The certificate store is available on all the computers in which Oracle Application Server is running.
- The Oracle HTTP Server is using HTTP port 80 and HTTPS port 443.
- The ORMI port is 12401 and the ORMIS port is 12701 for the Oracle Application Server instances.

Enabling SSL for HTTP Communication to Oracle HTTP Server

By default, the Oracle HTTP Server is configured with SSL and the SSL certificate store, which is located at *`ORACLE_HOME`*`\Apache\Apache\conf\ssl.wlt\default\`. The listen parameter in the *`ORACLE_HOME`*`\Apache\Apache\conf\ssl.conf` file points to the SSL port being used by the Oracle HTTP Server.

No configuration change is required for using the default certificate store that comes along with the installation.

Enabling SSL for RMI Communication to Oracle Application Server Instances

The Design Console communicates with EJBs deployed on the Oracle Application Server instances by using the ORMI protocol, which is unsecure. For using secured ORMIS protocol for communication between Oracle Application Server and the

Design Console, you must make modifications to both Oracle Application Server as well as the Design Console. The following sections provide information related to the configuration changes required for a successful SSL connection:

Configuring Oracle Application Server

The following sections explain the configuration changes required for Oracle Application Server:

> **See Also:** The "SSL Communication" section in *Oracle Containers for J2EE Security Guide* for more information about configuring ORMIS

Changes to server.xml

To enable ORMIS in an Oracle Application Server instance, you must ensure that `server.xml`, the Oracle Application Server configuration file, contains an `<rmi-config>` element that specifies the path to `rmi.xml`, the Oracle Application Server RMI configuration file. To do so:

1. Open the *ORACLE_HOME*`/j2ee/`*OC4J_INSTANCE*`/config/server.xml` file in a text editor.
2. Specify the path to `rmi.xml` as follows:

```
<rmi-config path="rmi_path" />
```

Because both the `server.xml` file and the `rmi.xml` file are typically in the *ORACLE_HOME*`/j2ee/`*OC4J_INSTANCE*`/config` directory, the typical value for rmi_path is `./rmi.xml`.

Changes to rmi.xml

To enable the server to use the ORMIS protocol as well as to specify the keystore to be used for SSL communication, you must make the following changes:

1. Open the *ORACLE_HOME*`/j2ee/`*OC4J_INSTANCE*`/config/rmi.xml` file in a text editor.
2. Modify the `rmi-server` element with a keystore value as follows:

```
<rmi-server ...  ssl-port="23943">
         …
         …
        <ssl-config
keystore="ORACLE_HOME\Apache\Apache\conf\ssl.wlt\default\ewallet.p12"
keystore-password="welcome" />
</rmi-server>
```

> **Note:** The default password for ewallet.p12 store is `welcome`. The password for the default certificate in the store is also `welcome`.

> **Note:** In case of a clustered setup, copy ewallet.p12 from the Web server to all nodes locally and specify the local path for the same.

Exporting Certificate

You must export the certificate from the default Oracle wallet *ORACLE_HOME*`\Apache\Apache\conf\ssl.wlt\default\ewallet.p12` for the

Design Console. This certificate is used for the Design Console to trust Oracle Application Server. To export the certificate, you first start Oracle Wallet Manager, as follows:

For Microsoft Windows:

Click **Start**, **Programs**, **Oracle-HOME_NAME**, **Integrated Management Tools**, and then click **Wallet Manager**.

For UNIX:

At the command line, go to *ORACLE_HOME*/bin/ and enter **owm**.

After you have started Oracle Wallet Manager, perform the following steps:

1. Open the *ORACLE_HOME*/Apache/Apache/conf/ssl.wlt/default/ directory by using Oracle Wallet Manager.
2. Enter the store password as **welcome** when prompted.
3. Right click `Certificate (Ready)` and click **Export User Certificate**.
4. Enter the file name as `server.cert` and save.

   This certificate is used by the Design Console to trust Oracle Application Server.

   See Also: The "Secure Sockets Layer" section in *Oracle Application Server Administrator's Guide* for more information about Oracle Wallet Manager

Changes to opmn.xml

You must make the following changes in the `opmn.xml` file:

1. Open the *ORACLE_HOME*\opmn\conf\opmn.xml file in a text editor.
2. Modify the following:

   ```
   <port id="rmis" range="12701-12800"/>
   ```

   to a single port usage as follows:

   ```
   <port id="rmis" range="12701"/>
   ```

   Note: The change in the port ID is mandatory to ensure that the ORMIS port is always unique.

3. Restart the corresponding Oracle Application Server instance.

   Note: For a clustered setup, all of these changes are required for all the nodes.

Configuring the Design Console

The following sections provide information about the changes required for the Design Console:

Changes to xlconfig.xml

By default, the Design Console uses the ORMI port to connect to Oracle Application Server and HTTP for connecting to the Oracle HTTP Server. In order to enable SSL

communication, you must configure the Design Console to use ORMIS and HTTPS connections. To do so:

1. Open the *OIM_DC_HOME*\xlclient\Config\xlconfig.xml file in a text editor.
2. Make the following modification:
   - Change

     <java.naming.provider.url>**ormi**://*SERVER_HOST*:**12401**</java.naming.provider.url>

     to

     <java.naming.provider.url>**ormis**://*SERVER_HOST*:**12701**</java.naming.provider.url>

     Note: For a clustered installation, ensure that you add the participating nodes with corresponding SSL port as comma separated values in the URL for `java.naming.provider.url`.

     ```
     <java.naming.provider.url> ormis://node1:12701,
     ormis://node2:12702</java.naming.provider.url>
     ```

   - Change

     <ApplicationURL>**http**://*SERVER_HOST*/xlWebApp/loginWorkflowRenderer.do</ApplicationURL>

     to

     <ApplicationURL>**https**://*SERVER_HOST*/xlWebApp/loginWorkflowRenderer.do</ApplicationURL>

     Note: It is assumed that 12401 is the ORMI port and 12701 is the ORMIS port of the Oracle Application Server instance. In addition, HTTP port is 80 and HTTPS port is 443 for Oracle HTTP Server. ORMI and ORMIS ports can be viewed from the Oracle Application Server Administrative Console.

     For more information, refer to *Oracle Containers for J2EE Configuration and Administration Guide*.

Configuring the Trust Store

By default, the Design Console uses the *OIM_DC_HOME*\java\lib\security\cacerts as the trust store for the SSL communication. The default password for the store is `changeit`. The server certificate must be imported to this store to make the Design Console trust Oracle Application Server. To configure the trust store:

1. Copy `server.cert` from Oracle Application Server to the Design Console at the following location:

   OIM_DC_HOME\java\lib\security

2. Import the Oracle Application Server certificate by using the following commands::

```
cd OIM_DC_HOME\java\lib\security
keytool -import -trustcacerts -alias oimserver1 -keystore cacerts -file
server.cert -storepass changeit -keypass welcome
```

Note: For a clustered installation, repeat the "Configuring the Trust Store" step for all the Oracle Application Server instances. When you import the certificate by using keytool, ensure that you use the unique alias for each Oracle Application Server instance in a cluster.

Note: This document describes the use of the default store, `ewallet.p12` for implementing SSL for the Design Console. Oracle recommends that for the use of certificate authority certificates for production implementation.

For more information refer to *Oracle Application Server Administrator's Guide*, *Oracle Containers for J2EE Security Guide*, and *Oracle HTTP Server Administrator's Guide*.

Removing the Design Console Installation

To remove the Design Console installation, perform the following steps:

1. Stop Oracle Identity Manager and the Design Console if they are running.
2. Stop all Oracle Identity Manager processes.
3. Delete the *OIM_DC_HOME* directory in which you installed the Design Console.

11

Installing and Configuring the Oracle Identity Manager Remote Manager

This chapter explains how to install Oracle Identity Manager Remote Manager. It contains the following sections:

Installing the Remote Manager on Microsoft Windows

Complete the following steps to install the Remote Manager on a Microsoft Windows host:

> **Important:** All Oracle Identity Manager components must be installed in different home directories. If you are installing the Remote Manager on a computer that is hosting another Oracle Identity Manager component (the server or the Design Console), specify an installation directory that has not been used.

1. Insert the Oracle Identity Manager Installation CD into your CD-ROM drive.
2. Using Windows Explorer, navigate to the installServer directory in the installation CD.
3. Double-click the setup_rm.exe file.
4. Choose a language from the list on the Installer page. The Welcome page is displayed.
5. On the Welcome page, click **Next**.
6. On the Target directory page, complete one of the following sub-steps:
   a. The default directory for Oracle Identity Manager products is C:\oracle. To install the Remote Manager in this directory, click **Next**.
   b. To install Remote Manager in a different directory, specify the path of the directory in the **Directory name** field, and then click **Next**.

> **Note:** If the directory path that you specified does not exist, then the Base Directory settings field is displayed. Click **OK**. The directory is automatically created. If you do not have write permission to create the default directory for Oracle Identity Manager, then a message is displayed informing you that the installer could not create the directory. Click **OK** to close the message, and then contact your system administrator to obtain the appropriate permissions.

7. Select either the JRE that is installed with Oracle Identity Manager or specify an existing JRE. Click **Next**. The Remote Manager Configuration page is displayed.
8. On the Remote Manager Configuration page, enter the appropriate information for the Remote Manager:
   a. Enter the service name. The default value is RManager.
   b. Use the default, prepopulated value of 12346 as the binding port.
   c. Use the default, prepopulated value of 12345 as the Remote Manager SSL port.
   d. Click **Next**.
9. On the Shortcut page, select (or deselect) the check boxes for the following shortcut options according to your preferences:
   a. Choose to create a shortcut for the Remote Manager on the desktop.
   b. Choose to create a shortcut for the Remote Manager on the Start Menu.

   Click **Next** when you are satisfied with the check box settings.
10. On the Summary page, review the configuration details, and then click **Install** to begin the installation.
11. After the installation has completed, click **Finish** on the Completed page to exit.

Installing the Remote Manager on UNIX or Linux

To install the Remote Manager on UNIX or Linux:

> **Note:** Before installing the Remote Manager, you must set the JAVA_Home variable to the certified JRE.

1. Insert the Oracle Identity Manager Installation CD into your CD-ROM drive.
2. From the File Manager, access the installServer directory in the installation CD.
3. Run the install_rm.sh file. The command-line installer starts.
4. Choose a language from the list by entering a number and then entering **0** to apply the language. The Welcome panel is displayed.
5. On the Welcome panel, enter **1** to move to the next panel. The Target directory panel is displayed.
6. On the Target directory panel, enter the path to the directory in which you want to install the Oracle Identity Manager Remote Manager. The default directory is /opt/oracle.
   - Enter **1** to move to the next panel.

- If the directory does not exist, you are asked to create it. Enter **y** to create the directory.

  Note: All Oracle Identity Manager components must be installed in different home directories. If you are installing the Remote Manager on a computer that is hosting an Oracle Identity Manager server, you must specify a unique installation directory.

7. Specify the JRE to use with Remote Manager:
   - Enter **1** to install the JRE included with Oracle Identity Manager.
   - Enter **2** to use an existing JRE at a specified location.

   After specifying the JRE, enter **0** to accept your selection and then enter **1** to move to the next panel.
8. On the Remote Manager Configuration panel, enter the Remote Manager configuration information:
   a. Enter the Service Name, or press **Enter** to accept the default value.
   b. Enter 12346 as the Remote Manager binding port.
   c. Enter 12345 as the Remote Manager SSL port.
   d. Enter **1** to move to the next panel.

      The Remote Manager installation summary panel is displayed.
9. Check the information.
   - Enter **2** to go back and make changes.
   - Enter **1** to start the installation.

     Oracle Remote Manager installs and the Post Install Summary panel is displayed.
10. Enter **3** to finish the Remote Manager installation.

Configuring the Remote Manager

The Remote Manager and Oracle Identity Manager server communicate using Secure Sockets Layer (SSL). If you are using Remote Manager, you must enable a trust relationship between your Oracle Identity Manager server and the Remote Manager. (The server must trust the Remote Manager certificate).

You also have the option to enable client-side authentication (where the Remote Manager checks the server's certificate). Import the Remote Manager's certificate into your Oracle Identity Manager server's keystore and make it trusted. For client-side authentication, import the certificate for your Oracle Identity Manager server into the keystore for your Remote Manager, then make that certificate trusted. You must also manually edit the configuration file associated with the server, and depending on the options you selected during Remote Manager installation, the Remote Manager configuration file as well.

Changing the Remote Manager Keystore Passwords

During installation, the password for the Remote Manager keystore is set to `xellerate`. Oracle recommends that for changing the keystore passwords for all production installations.

To change the keystore passwords, you must change the storepass of .xlkeystore and the keypass of the xell entry in .xlkeystore—and these two values must be identical. Use the keytool utility and the following steps to change the keystore passwords:

1. Open a command prompt on the Oracle Identity Manager host computer.
2. Navigate to the *OIM_RM_HOME*`\xellerate\config` directory.
3. Run the keytool utility with the following options to change the storepass:

   ```
   JAVA_HOME\jre\bin\keytool -storepasswd -new new_password -storepass xellerate
   -keystore .xlkeystore -storetype JKS
   ```

4. Run the keytool utility with the following options to change the keypass of the xell entry in .xlkeystore:

   ```
   JAVA_HOME\jre\bin\keytool -keypasswd -alias xell -keypass xellerate
   -new new_password -keystore .xlkeystore -storepass xellerate
   ```

   JAVA_HOME represents the location of the Java installation associated with the Remote Manager installation.
5. Open *OIM_RM_HOME*`\xlremote\config\xlconfig.xml` in a text editor: .
6. Edit the <RMSecurity>.<KeyStore> to specify the keystore password as follows:
   - Change the password tag to `encrypted="false"`.
   - Enter the password, for example:

     ```
     <RMSecurity>
     <KeyStore>
     <Location>.xlkeystore</Location>
     <Password encrypted="false">new_password</Password>
     <Type>JKS</Type>
     <Provider>sun.security.provider.Sun</Provider>
     </KeyStore>
     ```

   Note: If you are using client-side authentication for the Remote Manager, enter the Oracle Identity Manager's keystore password in the <RMSecurity>.<TrustStore> section of *OIM_RM_HOME*`\xlremote\config\xlconfig.xml` as follows:

   ```
   <TrustStore>
   <Location>.xlkeystore</Location>
   <Password encrypted="false">OIM_Server_keystore_password</Password>
   <Type>JKS</Type>
   <Provider>sun.security.provider.Sun</Provider>
   </TrustStore>
   ```

7. Save and close the xlconfig.xml file.
8. Restart the Remote Manager.
9. Open *OIM_HOME*`\xellerate\config\xlconfig.xml` in a text editor.

10. Edit the <RMSecurity>.<TrustStore> to specify the new Remote Manager keystore password as follows:
    - Change the password tag to `encrypted="false"`.
    - Enter the password (in the clear), for example:

```
<TrustStore>
<Location>.xlkeystore</Location>
<Password encrypted="false">new_password</Password>
<Type>JKS</Type>
<Provider>sun.security.provider.Sun</Provider>
</TrustStore>
```

11. Save and close the xlconfig.xml file, then restart Oracle Identity Manager.

Trusting the Remote Manager Certificate

To establish a trust relationship between Oracle Identity Manager and the Remote Manager:

1. Copy the Remote Manager certificate to the server computer. On the Remote Manager computer, locate the file *OIM_RM_HOME*`\xlremote\config\xlserver.cert` and copy it to the server computer.

   Note: The server certificate in *OIM_HOME*`\config` is also named `xlserver.cert`. Ensure that you do not overwrite that certificate.

2. Open a command prompt on the server computer.
3. To import the certificate by using the keytool utility, use the following command:

```
JAVA_HOME\jre\bin\keytool -import -alias rm_trusted_cert -file
RM_cert_location\xlserver.cert -trustcacerts -keystore
XL_HOME\xellerate\config\.xlkeystore -storepass xellerate
```

   JAVA_HOME is the location of the Java directory for your application server, the value of *alias* is an arbitrary name for the certificate in the store, and *RM_cert_location* is the location where you copied the certificate.

   Note: If you changed the keystore password, substitute that for xellerate for the value of the storepass variable.

4. Enter **Y** at the prompt to trust the certificate.
5. Open *OIM_HOME*`\xellerate\config\xlconfig.xml` in a text editor.
6. Locate the <RMIOverSSL> property and set it to `true`, for example:

```
<RMIOverSSL>true</RMIOverSSL>
```

7. Locate the <KeyManagerFactory> property and set the value to `SUNX509`. For example:

```
<KeyManagerFactory>SUNX509</KeyManagerFactory>
```

8. Save the file.

9. Restart Oracle Identity Manager.

Using Your Own Certificate

To configure the Remote Manager by using your own certificate on the Remote Manager system:

1. Import your custom key in a new keystore (new_keystore_name) other than .xlkeystore. Remember the password (new_keystore_pwd) that you use for the new keystore.
2. Copy this new keystore to the *OIM_RM_HOME*\xlremote\config\ directory.
3. Open *OIM_RM_HOME*\xlremote\config\xlconfig.xml in a text editor.
4. Locate the <RMSecurity> tag and change the value in the <Location> and <Password> tags as follows:

```
<KeyStore>
     <Location>new_keystore_name</Location>
     <Password encrypted="false">new_keystore_pwd</Password>
     <Type>JKS</Type>
     <Provider>sun.security.provider.Sun</Provider>
</KeyStore>
```

5. Restart the Remote Manager server, and open the xlconfig.xml file to ensure that the password for the new keystore was encrypted.

To configure the Remote Manager by using your own certificate on the Oracle Identity Manager server:

1. Import the same certificate key used in the Remote Manager system to a new keystore (new_svrkeystore_name) other than .xlkeystore. Remember the password (new_svrkeystor_pwd) that you use for the new keystore.
2. Copy this new keystore to the *OIM_HOME*\xellerate\config directory.
3. Open *OIM_HOME*\xellerate\config\xlconfig.xml in a text editor.
4. Locate the <RMSecurity> tag and change the value in the <Location> and <Password> tags as follows:

```
<TrustStore>
   <Location>new_svrkeystore_name</Location>
   <Password encrypted="false">new_svrkeystor_pwd</Password>
   <Type>JKS</Type>
   <Provider>sun.security.provider.Sun</Provider>
</TrustStore>
```

5. Restart Oracle Identity Manager and open the xlconfig.xml file to ensure that the password for the new keystore is encrypted.

Enabling Client-Side Authentication for Remote Manager

To enable client-side authentication:

1. On the computer hosting the Remote Manager, open *OIM_RM_HOME*\xlremote\config\xlconfig.xml in a text editor.
2. Set the <ClientAuth> property to true, for example:

```
<ClientAuth>true</ClientAuth>
```

3. Ensure the <RMIOverSSL> property is set to true, for example:

```
<RMIOverSSL>true</RMIOverSSL>
```

4. Locate the <KeyManagerFactory> property and set the value to `SUNX509`. For example:

   ```
   <KeyManagerFactory>SUNX509</KeyManagerFactory>
   ```

5. Save the file.
6. Copy the server certificate to the Remote Manager computer. On the server computer, locate the file *OIM_HOME*`\xellerate\config\xlserver.cert` and copy it to the Remote Manager computer.

   Note: The Remote Manager certificate is also named xlserver.cert. Ensure that you do not overwrite that certificate.

7. Open a command prompt on the Remote Manager computer.
8. Import the certificate by using the following keytool command:

   ```
   JAVA_HOME\jre\bin\keytool -import -alias trusted_server_cert -file
   server_cert_location\xlserver.cert -trustcacerts -keystore
   XL_RM_HOME\xlremote\config\.xlkeystore -storepass xellerate
   ```

   JAVA_HOME is the location of the Java directory for your Remote Manager, the value of *alias* is an arbitrary name for the certificate in the store, *OIM_RM_HOME* is the home directory for the Remote Manager, and *server_cert_location* is the location to which you copied the server certificate.

   Note: If you changed the keystore password, substitute that value for xellerate, which is the default value of the storepass variable.

9. Enter **Y** at the prompt to trust the certificate.
10. Restart the Remote Manager.

Starting the Remote Manager

Use the following script to start the Remote Manager:

- On Microsoft Windows:

  OIM_RM_HOME`\xlremote\remotemanager.bat`

- On UNIX:

  OIM_RM_HOME`/xlremote/remotemanager.sh`

Removing the Remote Manager Installation

To remove the Remote Manager installation, perform the following steps:

1. Stop Oracle Identity Manager and the Remote Manager if they are running.
2. Stop all Oracle Identity Manager processes.
3. Delete the *OIM_RM_HOME* directory in which you installed the Remote Manager.

12

Troubleshooting the Oracle Identity Manager Installation

This chapter describes problems that can occur during the Oracle Identity Manager Installation and contains the following topics:

> **Note:** You can use the Diagnostic Dashboard tool for assistance when you troubleshoot the Oracle Identity Manager Installation. See *Oracle Identity Manager Administrative and User Console Guide* for detailed information.

Task Scheduler fails in a Clustered Installation

The Task Scheduler fails to work properly when the cluster members (computers that are part of the cluster) have different settings on their system clocks. Oracle highly recommends that the system clocks for all cluster members be synchronized within a second of each other.

A

Java 2 Security for Oracle Application Server

Note: The application might fail to start because of syntax errors in the policy files.

Be careful when you edit the policy files. Oracle recommends that you use the policy tool provided by the JDK for editing the policy files. The tool is available in the following directory:

JAVA_HOME/jre/bin/policytool

To enable Java 2 Security for Oracle Identity Manager running on Oracle Application Server:

1. Modify the Oracle Application Server run configuration and add the `-Djava.security.manager` as a JVM option. This change must be done in `$OC4J_HOME/opmn/conf/opmn.xml`.
2. Add the following option to Oracle Application Server:

   ```
   -Djava.security.manager
   ```

   This option enables the Java 2 Security manager.
3. Check if the *$ORACLE_HOME*`/j2ee/home/config/java2.policy` file exists. If it exists, then edit it and add the Java 2 Security permissions listed in the "Policy File" section on page A-1. If the `java2.policy` file does not exist, then you have to create it.

Policy File

Perform the following in the `java2.policy` file:

Note:

- The instructions to change the code in the policy file are given in comments, which are in bold font.

- This `java2.policy` example is for Windows installation. For UNIX, ensure that you change `\\` between the directories name to `/` in every permission `java.io.FilePermission` property.

- Make sure to change the multicast IP `231.184.202.110` in this example to reflect the multicast IP address of the Oracle Identity Manager installation. You can find the Oracle Identity Manager multicast IP address in `xlconfig.xml`.

- You must update the path to the correct value for the location where GTC-RECON connector files are located. This example uses `C:\\file1\\file1` for the location of these files.

1. Find the following:

```
grant { permission java.util.PropertyPermission
"javax.xml.parsers.DocumentBuilderFactory" , "read";
};
```

Add the following to the preceding code:

```
grant { permission java.util.PropertyPermission
"javax.xml.parsers.DocumentBuilderFactory", "write";
};
```

2. Find **`/*Default Grants copied from the JDK default system policy*/`** and add the following code to the grant:

```
//Added for OIM
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "*", "write";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.net.SocketPermission "*", "connect";
permission java.lang.RuntimePermission "accessClassInPackage.*";
permission javax.management.MBeanServerPermission "findMBeanServer";
permission javax.security.auth.AuthPermission "createLoginContext.*";

//Added for AQ
permission java.lang.RuntimePermission "accessDeclaredMembers";

// For Nexaweb
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "setContextClassLoader";
permission java.util.PropertyPermission "nexaweb.logs", "read,write";
permission java.util.PropertyPermission"sun.net.client.defaultConnectTimeout",
"read,write";
permission java.util.PropertyPermission "sun.net.client.defaultReadTimeout",
"read,write";
permission java.lang.RuntimePermission "loadLibrary.*";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.net.SocketPermission "*", "connect";
permission java.io.FilePermission "<<ALL FILES>>", "read";
permission java.lang.RuntimePermission "modifyThreadGroup";
permission oracle.oc4j.security.OC4JRuntimePermission "oracle.oc4j.OC4JOnly";
permission javax.management.MBeanPermission
```

```
"oracle.oc4j.admin.jmx.server.mbeans.model.DefaultModelMBeanImpl#-", "*";

//Change this to the original directory where logs are being geting created
//If logs are getting created in more then one directory ensure that you have
two entries for them here.
permission java.io.FilePermission "${oracle.home}\\opmn\\logs\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\logs\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\velocity.log",
"read,write,delete";

/*
* permission java.io.FilePermission "C:\\files\\file1\\-", "read,write,delete";
* property has been added for the path of directory where files are kept for
* the GTC-RECON connector. Update the path to the correct value prior to
* running the server.
*/
permission java.io.FilePermission "C:\\files\\file1\\-", "read,write,delete";
```

3. In Custom Application Permissions, append the following code:

```
// Java code and extensions
// Trust java extensions
grant codeBase "file:${java.home}/lib/ext/-" {
permission java.security.AllPermission;
};

/*grant codeBase "file:${XL.HomeDir}/logs/-" {
permission java.security.AllPermission;
  };
*/

// Trust core java code
grant codeBase "file:${java.home}/lib/*" {
permission java.security.AllPermission;
};

// For java.home pointing to the JDK jre directory
grant codeBase "file:${java.home}/jre/lib/-" {
permission java.security.AllPermission;
};

// Grant All permissions to nexaweb commons jar file to be loaded from
grant codeBase "file:${oracle.home}/j2ee/home/applib/nexaweb-common.jar" {
permission java.security.AllPermission;
};

// OIM codebase permissions
grant codeBase "file:${oracle.home}/j2ee/home/applications/Xellerate/-" {

// File permissions
// Need read, write, and delete permissions on $OIM_HOME/config folder
// to read various config files, write the
// xlconfig.xml.{0,1,2..} files upon re-encryption and delete
// the last xlconfig.xml if the numbers go above 9.
permission java.io.FilePermission "${XL.HomeDir}\\config\\-",
 "read, write, delete";
permission java.io.FilePermission "${XL.HomeDir}\\-", "read";

// Need read,write,delete permissions to generate adapter java
```

```
// code, delete the .class file when the adapter is loaded into
// the database
permission java.io.FilePermission "${XL.HomeDir}\\adapters\\-",
     "read,write,delete";

// This is required by the connectors and connector installer
permission java.io.FilePermission
"${XL.HomeDir}\\ConnectorDefaultDirectory\\-",
  "read,write,delete";
permission java.io.FilePermission
"${XL.HomeDir}\\adapters\\connectorResources\\-",
  "read,write,delete";

// Read Globalization resource bundle files for various
// locales
permission java.io.FilePermission
"${XL.HomeDir}\\adapters\\customResources\\-", "read";

// Read code from "JavaTasks", "ScheduleTask",
// "ThirdParty", "EventHandlers" folder
permission java.io.FilePermission "${XL.HomeDir}\\EventHandlers\\-", "read";
permission java.io.FilePermission "${XL.HomeDir}\\JavaTasks\\-", "read";
permission java.io.FilePermission "${XL.HomeDir}\\ScheduleTask\\-", "read";
permission java.io.FilePermission "${XL.HomeDir}\\ThirdParty\\-", "read";

// Required by the Generic Technology connector
permission java.io.FilePermission "${XL.HomeDir}\\GTC\\-", "read";

// Server needs read permissions on Nexaweb home directory
//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Read permissions on the "application-deployments" folder, the OIM deploy
// directory
permission java.io.FilePermission
"${oracle.home}\\j2ee\\home\\application-deployments\\Xellerate\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\home\\applications\\Xellerate\\-", "read,write,delete";

// OIM server invokes the java compiler. You need "execute"
// permissions on all files.
permission java.io.FilePermission "<<ALL FILES>>", "execute";

// Socket permissions
// Basically you allow all permissions on nonprivileged sockets
// The multicast address should be the same as the one in
// xlconfig.xml for javagroups communication
        permission java.net.SocketPermission "*",
        "connect,listen,resolve,accept";
        permission java.net.SocketPermission "231.184.202.110",
        "connect,accept";

// Property permissions
// Read and write OIM properties
// Read XL.*, java.* and log4j.* properties
permission java.util.PropertyPermission "XL.*", "read,write";
permission java.util.PropertyPermission "*", "read, write";
permission java.util.PropertyPermission "java.*", "read";
```

```
permission java.util.PropertyPermission "log4j.", "read";
permission java.util.PropertyPermission "user.dir", "read";

// Runtime permissions
// OIM server needs permissions to create its own class loader,
// get the class loader, modify threads and register shutdown
// hooks
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "modifyThread";
permission java.lang.RuntimePermission "modifyThreadGroup";
permission java.lang.RuntimePermission "shutdownHooks";

// OIM server needs runtime permissions to generate and load
// classes in the below specified packages. Also access the
// declared members of a class.
permission java.lang.RuntimePermission
"defineClassInPackage.com.thortech.xl.adapterGlue.ScheduleItemEvents";
permission java.lang.RuntimePermission
"defineClassInPackage.com.thortech.xl.dataobj.rulegenerators";
permission java.lang.RuntimePermission
"defineClassInPackage.com.thortech.xl.adapterGlue";
permission java.lang.RuntimePermission "accessDeclaredMembers";

// Reflection permissions
// Give permissions to access and invoke fields/methods from
// reflected classes.
        permission java.lang.reflect.ReflectPermission
                "suppressAccessChecks";

// Security permissions for OIM server
permission java.security.SecurityPermission "*";
permission javax.security.auth.AuthPermission "doAs";
permission javax.security.auth.AuthPermission "doPrivileged";
permission javax.security.auth.AuthPermission "getSubject";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission "createLoginContext";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission javax.security.auth.AuthPermission "getLoginConfiguration";
permission javax.security.auth.AuthPermission "setLoginConfiguration";

// SSL permission (for remote manager)
permission javax.net.ssl.SSLPermission  "getSSLSessionContext";
permission java.net.SocketPermission "*:1024-", "listen";
permission java.util.logging.LoggingPermission "control";
permission java.lang.RuntimePermission "enableContextClassLoaderOverride";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.io.SerializablePermission "enableSubstitution";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission javax.security.auth.AuthPermission "getPolicy";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\Xellerate.err",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};

// Nexaweb server codebase permissions
```

```
grant codeBase "file:${oracle.home}/j2ee/home/applications/Nexaweb/-" {
// File permissions
permission java.io.FilePermission "${user.home}", "read, write";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\home\\application-deployments\\Nexaweb\\-",
"read,write,delete";

//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Property permissions
permission java.util.PropertyPermission "*", "read,write";

// Runtime permissions
// Nexaweb server needs permissions to create its own class loader,
// get the class loader etc.
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "setContextClassLoader";
permission java.lang.RuntimePermission  "setFactory";

// Nexaweb server security permissions to load the Cryptix
// extension
        permission java.security.SecurityPermission
        "insertProvider.Cryptix";

// Socket permissions
// Permissions on all non-privileged ports.
        permission java.net.SocketPermission "*:1024-",
                "listen, connect, resolve";

// Security permissions
permission javax.security.auth.AuthPermission "doAs";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission "createLoginContext";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.util.logging.LoggingPermission "control";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.io.SerializablePermission "enableSubstitution";
permission javax.security.auth.AuthPermission "getPolicy";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};

// The following are permissions given to codebase in the OIM server
// directory
grant codeBase "file:${XL.HomeDir}/-" {
// File permissions
permission java.io.FilePermission "${XL.HomeDir}\\config\\-",
      "read";
permission java.io.FilePermission "${XL.HomeDir}\\JavaTasks\\-",
      "read";
permission java.io.FilePermission
      "${XL.HomeDir}\\ScheduleTasks\\-", "read";
permission java.io.FilePermission
```

```
      "${XL.HomeDir}\\ThirdParty\\-", "read";
permission java.io.FilePermission
      "${XL.HomeDir}\\adapters\\-", "read,write,delete";

//permission java.io.FilePermission "${nexaweb.home}\\-", "read";
// Socket permissions
       permission java.net.SocketPermission "*", "listen";

// Property permissions
// Read XL.* and log4j.* properties
       permission java.util.PropertyPermission "XL.*", "read";
       permission java.util.PropertyPermission "log*", "read";

// Security permissions
permission javax.security.auth.AuthPermission "doAs";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission "createLoginContext";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.io.SerializablePermission "enableSubstitution";
permission java.util.logging.LoggingPermission "control";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.security.SecurityPermission "*";
permission javax.security.auth.AuthPermission
               "getLoginConfiguration";
permission javax.security.auth.AuthPermission
               "getPolicy";
permission javax.security.auth.AuthPermission
               "setLoginConfiguration";
permission java.security.SecurityPermission
       "insertProvider.Cryptix";

// Socket permissions
// Permissions on all non-privileged ports.
permission java.net.SocketPermission "*:1024-","listen, connect, resolve";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};
```

Policy File

The following is the sample `java2.policy` file after Oracle Identity Manager policy has been added:

```
/*
 * Standard policy file for Oracle Application Server
 *
 *    When this file is in use the System property ${oracle.home} must
 *    be set to $ORACLE_HOME or to the value of $ORACLE_HOME.
 *
 *    When this file is in use via OPMN the System property
 *    ${oracle.oc4j.instancename}
 *    is used to identify the instance-level connector jars.
 *
 *    This file grants AllPermission to "oc4j code"
```

```
 *    oc4j code is code used either directly or indirectly by the app server
 *    itself. Including code generated for ejb wrappers.
 *    See oc4j.jar!boot.xml for a complete list. Currently this file
 *    only lists jars that need permissions. Others can be
 *    added if neccessary.
 *
 *    In a future release the grants will be refined so that
 *    only the Permissions actually needed by Oracle Application Server
 *    code will be granted.
 *
 *    Calls to accessController.doPrivileged have been added to Oracle
 *    Application Server with the intention that the application code only
 *    be granted the Permissions needed by actions it performs directly.
 *    It should not be granted Permissions required by J2EE
 *    operations.
 *
 *    For example if a Servlet (or jsp) forwards to a .jsp it does not
 *    need Permission to read and compile the .jsp.  Similarly the
 *    application code associated with an ejb that specifies container
 *    managed persistence does not need Permission to create a socket
 *    talking to the database holding the underlying data. But an EJB
 *    using bean managed persistence does need such Permission.
 */

grant codebase "file:${oracle.home}/j2ee/home/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/lib/*" {
    permission java.security.AllPermission;
};


grant codebase "file:${oracle.home}/jlib/-" {
    permission java.security.AllPermission;
};


grant codebase "file:${oracle.home}/bc4j/jlib/*" {
    permission java.security.AllPermission;
};

grant codeBase "file:${oracle.home}/toplink/jlib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/dms/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/diagnostics/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/jdbc/lib/ojdbc14dms.jar" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/dbjava/lib/*" {
    permission java.security.AllPermission;
```

```
};

grant codebase "file:${oracle.home}/sqlj/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/javacache/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/uddi/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/xdk/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/opmn/lib/*" {
    permission java.security.AllPermission;
};


grant codebase "file:${oracle.home}/webservices/lib/*" {
    permission java.security.AllPermission;
};



grant codeBase "file:${oracle.home}/javavm/lib/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/jsp/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/lib/*" {
    permission java.security.AllPermission;
};

/** EJB skeleton/tie & BCEL proxy support **/
grant codeBase "file:generated/by/proxy" {
    permission java.security.AllPermission;
};

grant codeBase
"file://generated/by/oracle.j2ee.connector.proxy.BCELProxyClassLoader" {
    permission java.security.AllPermission;
};

* Miscellaneous grants to jars distributed as part of oc4j that might be used
* in various ways
*/
grant codebase
"file:${oracle.home}/j2ee/home/connectors/OracleASjms/OracleASjms/gjra.jar" {
    permission java.security.AllPermission;
};

grant codebase
```

```
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/connectors/OracleASjms/Oracl
eASjms/gjra.jar" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/home/connectors/datasources/datasources/datasources.jar"
{
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/connectors/datasources/datas
ources/datasources.jar" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/jsp/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/jsp/lib/taglib/ojsputil.jar" {
    permission java.security.AllPermission;
};

/* GRANTS TO DEFAULT APPLICATIONS */

grant codebase "file:${oracle.home}/j2ee/home/application-deployments/ascontrol/-"
{
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/asco
ntrol/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/ascontrol/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/ascontrol/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/application-deployments/default/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/defa
ult/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/default/-" {
    permission java.security.AllPermission;
};
```

```
grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/default/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/application-deployments/javasso/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/java
sso/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/javasso/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/javasso/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/application-deployments/usermbean/-"
{
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/user
mbean/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/usermbean/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/usermbean/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/application-deployments/admin_ejb/-"
{
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/admi
n_ejb/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/admin_ejb.jar" {
    permission java.security.AllPermission;
};
```

```
grant codebase "file:${oracle.home}/j2ee/home/applications/admin_ejb/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/admin_ejb/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/jmsrouter-ejb.jar" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/jmsrouter" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/home/application-deployments/JMXSoapAdapter-web/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/JMXS
oapAdapter-web/-" {
    permission java.security.AllPermission;
};


grant { permission java.util.PropertyPermission "j2ee.home", "read"; } ;
grant { permission java.util.PropertyPermission "java.home", "read"; } ;
grant { permission java.util.PropertyPermission "javax.xml.soap.SOAPFactory",
"read"; } ;
grant { permission java.util.PropertyPermission "javax.activation.debug" , "read";
} ;
grant { permission java.util.PropertyPermission
"javax.xml.parsers.DocumentBuilderFactory" , "read"; } ;


//Added for GTC
grant { permission java.util.PropertyPermission
"javax.xml.parsers.DocumentBuilderFactory", "write"; };

grant { permission java.util.PropertyPermission "javax.xml.soap.MessageFactory" ,
"read"; } ;
grant { permission java.util.PropertyPermission "jdbc.nontx.autocommit" , "read";
} ;
grant { permission java.util.PropertyPermission "mail.URLName.dontencode" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oc4j.jmx.event.interval" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oc4j.jmx.heartbeat.interval" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.defaultNChar" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.DMSStatementMetrics"
, "read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.J2EE13Compliant" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.TcpNoDelay" , "read";
```

```
} ;
grant { permission java.util.PropertyPermission
"oracle.jdbc.useFetchSizeWithLongColumn" , "read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.V8Compatible" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jserver.version" , "read";
} ;
grant { permission java.util.PropertyPermission "oracle.xml.parser.debugmode" ,
"read"; } ;
grant { permission java.util.PropertyPermission
"oracle.xml.parser.default.character.set" , "read"; } ;
grant { permission java.util.PropertyPermission "oracle.xml.xslt.jdwp", "read"; };
grant { permission java.util.PropertyPermission "orasaaj.soapversion" , "read"; }
;
grant { permission java.util.PropertyPermission "org.apache.commons.logging.Log" ,
"read"; } ;
grant { permission java.util.PropertyPermission
"org.apache.commons.logging.LogFactory" , "read"; } ;
grant { permission java.util.PropertyPermission "PersistenceManagerDebug" ,
"read"; } ;
grant { permission java.util.PropertyPermission "pro.debug" , "read"; } ;
grant { permission java.util.PropertyPermission "sqlj.runtime" , "read"; } ;
grant { permission java.util.PropertyPermission "transaction.debug" , "read"; } ;
grant { permission java.util.PropertyPermission "user.home" , "read"; } ;
grant { permission java.util.PropertyPermission "user.name" , "read"; } ;
grant { permission java.util.PropertyPermission "rmi.verbose" , "read"; } ;
grant { permission java.util.PropertyPermission "AssociateUserToThread", "read";
};
grant { permission java.util.PropertyPermission
"toplink.cts.collection.checkParameters", "read"; };
grant { permission java.util.PropertyPermission "AllowZeroInPK", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.Modules", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.Nagle", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.cookies.hosts.accept",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.cookies.hosts.reject",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.cookies.save", "read";
};
grant { permission java.util.PropertyPermission "HTTPClient.deferStreamed",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.disableKeepAlives",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.disable_pipelining",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.dontChunkRequests",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.dontTimeoutRespBody",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.forceHTTP_1.0",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.log.level", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.nonProxyHosts",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.socket.idleTimeout",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.socksHost", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.socksPort", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.socksVersion", "read";
};
```

```
grant { permission java.util.PropertyPermission "JavaClass.debug", "read"; };
grant { permission java.util.PropertyPermission "LoadBalanceOnLookup", "read"; };
grant { permission java.util.PropertyPermission "SQLLog", "read"; };
grant { permission java.util.PropertyPermission "USE_JAAS", "read"; };
grant { permission java.util.PropertyPermission "com.sun.enterprise.home", "read";
};
grant { permission java.util.PropertyPermission
"customFinderMethod.noLazyLoading", "read"; };
grant { permission java.util.PropertyPermission "debug", "read"; };
grant { permission java.util.PropertyPermission "default.cmp.pm", "read"; };
grant { permission java.util.PropertyPermission "ejb.debug.verbose", "read"; };
grant { permission java.util.PropertyPermission "findByPrimaryKey.noLazyLoading",
"read"; };
grant { permission java.util.PropertyPermission "http.nonProxyHosts", "read"; };
grant { permission java.util.PropertyPermission "http.proxyHost", "read"; };
grant { permission java.util.PropertyPermission "http.proxyPort", "read"; };
grant { permission java.util.PropertyPermission "java.ext.dirs", "read"; };
grant { permission java.util.PropertyPermission "java.class.path", "read"; };
grant { permission java.util.PropertyPermission
"javax.xml.parsers.SAXParserFactory", "read"; };
grant { permission java.util.PropertyPermission "jca.connection.debug", "read"; };
grant { permission java.util.PropertyPermission "log4j.configDebug", "read"; };
grant { permission java.util.PropertyPermission "log4j.configuration", "read"; };
grant { permission java.util.PropertyPermission "log4j.debug", "read"; };
grant { permission java.util.PropertyPermission "log4j.defaultInitOverride",
"read"; };
grant { permission java.util.PropertyPermission "log4j.disable", "read"; };
grant { permission java.util.PropertyPermission "log4j.disableOverride", "read";
};
grant { permission java.util.PropertyPermission "oneToOneJoin", "read"; };
grant { permission java.util.PropertyPermission "sun.boot.class.path", "read"; };
grant { permission java.util.PropertyPermission "toplink.changePolicy", "read"; };
grant { permission java.util.PropertyPermission
"toplink.cts.collection.checkParameters", "read"; };
grant { permission java.util.PropertyPermission
"toplink.cts.collection.checkTransaction", "read"; };
grant { permission java.util.PropertyPermission
"toplink.defaultmapping.dbTableGenSetting", "read"; };
grant { permission java.util.PropertyPermission
"toplink.defaultmapping.useExtendedTableNames", "read"; };
grant { permission java.util.PropertyPermission "toplink.log.destination", "read";
};
grant { permission java.util.PropertyPermission "toplink.log.level", "read"; };
grant { permission java.util.PropertyPermission "toplink.xml.platform", "read"; };
grant { permission java.util.PropertyPermission "upload.buflen", "read"; };
grant { permission java.util.PropertyPermission "user.dir", "read"; };
grant { permission java.util.PropertyPermission
"javax.xml.soap.SOAPConnectionFactory", "read";};
grant { permission java.util.PropertyPermission "HTTPClient.socket.idleTimeout",
"write";};

/* JDK  */

grant codebase "file:${java.home}/../lib/tools.jar" {
    permission java.security.AllPermission;
};

grant codeBase "file:${java.home}/lib/ext/*" {
permission java.security.AllPermission;
};
```

```
/* Default Grants copied from the JDK default system policy. */

grant {
// "standard" properties that can be read by anyone.

permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.vendor", "read";
permission java.util.PropertyPermission "java.vendor.url", "read";
permission java.util.PropertyPermission "java.class.version", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "file.separator", "read";
permission java.util.PropertyPermission "path.separator", "read";
permission java.util.PropertyPermission "line.separator", "read";

permission java.util.PropertyPermission "java.specification.version", "read";
permission java.util.PropertyPermission "java.specification.vendor", "read";
permission java.util.PropertyPermission "java.specification.name", "read";

permission java.util.PropertyPermission "java.vm.specification.version", "read";
permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
permission java.util.PropertyPermission "java.vm.specification.name", "read";
permission java.util.PropertyPermission "java.vm.version", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "java.vm.name", "read";

/* The following are granted by the default jdk policy but are considered
* unsafe and are omitted by this policy file */

// permission java.lang.RuntimePermission "stopThread";
// permission java.net.SocketPermission "localhost:1024-", "listen";

// Added for Oracle Identity Manager
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "*", "write";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.net.SocketPermission "*", "connect";
permission java.lang.RuntimePermission "accessClassInPackage.*";
permission javax.management.MBeanServerPermission "findMBeanServer";
permission javax.security.auth.AuthPermission "createLoginContext.*";

//Added for AQ
permission java.lang.RuntimePermission "accessDeclaredMembers";

// For Nexaweb
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "setContextClassLoader";
permission java.util.PropertyPermission "nexaweb.logs", "read,write";
permission java.util.PropertyPermission
"sun.net.client.defaultConnectTimeout", "read,write";
permission java.util.PropertyPermission "sun.net.client.defaultReadTimeout",
"read,write";
permission java.lang.RuntimePermission "loadLibrary.*";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.net.SocketPermission    "*", "connect";
permission java.io.FilePermission       "<<ALL FILES>>", "read";
permission java.lang.RuntimePermission   "modifyThreadGroup";
permission oracle.oc4j.security.OC4JRuntimePermission "oracle.oc4j.OC4JOnly";
```

```
permission javax.management.MBeanPermission
"oracle.oc4j.admin.jmx.server.mbeans.model.DefaultModelMBeanImpl#-", "*";

//Change this to the original directory where logs are being created
//If logs are getting created in more then one directory ensure that you have two
entries for them here.
permission java.io.FilePermission "${oracle.home}\\opmn\\logs\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\logs\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\velocity.log",
"read,write,delete";

/*
*  permission java.io.FilePermission "C:\\files\\file1\\-", "read,write,delete";
*  property has been added for the path of directory where files are kept for
*  GTC-RECON connector. Update the path to correct value prior to runnung the
*  server.
*/
permission java.io.FilePermission "C:\\files\\file1\\-", "read,write,delete";

};

/**
** Add Custom Application Permission Grants Below
**/
// Java code and extensions
// Trust java extensions
grant codeBase "file:${java.home}/lib/ext/-" {
permission java.security.AllPermission;
};

/*grant codeBase "file:${XL.HomeDir}/logs/-" {
permission java.security.AllPermission;
  };
*/

// Trust core java code
grant codeBase "file:${java.home}/lib/*" {
permission java.security.AllPermission;
};

// For java.home pointing to the JDK jre directory
grant codeBase "file:${java.home}/jre/lib/-" {
permission java.security.AllPermission;
};


// Grant All permissions to nexaweb commons jar file to be loaded from
grant codeBase "file:${oracle.home}/j2ee/home/applib/nexaweb-common.jar" {
permission java.security.AllPermission;
};

// OIM codebase permissions
grant codeBase "file:${oracle.home}/j2ee/home/applications/Xellerate/-" {

// File permissions
// Need read,write,delete permissions on $OIM_HOME/config folder
// to read various config files, write the
// xlconfig.xml.{0,1,2..} files upon re-encryption and delete
```

```
// the last xlconfig.xml if the numbers go above 9.
permission java.io.FilePermission "${XL.HomeDir}\\config\\-",
 "read, write, delete";
permission java.io.FilePermission "${XL.HomeDir}\\-", "read";

// Need read,write,delete permissions to generate adapter java
// code, delete the .class file when the adapter is loaded into
// the database
permission java.io.FilePermission "${XL.HomeDir}\\adapters\\-",
        "read,write,delete";

// This is required by the connectors and connector installer
permission java.io.FilePermission "${XL.HomeDir}\\ConnectorDefaultDirectory\\-",
          "read,write,delete";
permission java.io.FilePermission
"${XL.HomeDir}\\adapters\\connectorResources\\-",
          "read,write,delete";

// Read Globalization resource bundle files for various
// locales
        permission java.io.FilePermission
        "${XL.HomeDir}\\adapters\\customResources\\-", "read";

// Read code from "JavaTasks", "ScheduleTask",
// "ThirdParty", "EventHandlers" folder
permission java.io.FilePermission
       "${XL.HomeDir}\\EventHandlers\\-", "read";
permission java.io.FilePermission
       "${XL.HomeDir}\\JavaTasks\\-", "read";
permission java.io.FilePermission
       "${XL.HomeDir}\\ScheduleTask\\-", "read";
permission java.io.FilePermission
       "${XL.HomeDir}\\ThirdParty\\-", "read";

// Required by the Generic Technology connector
        permission java.io.FilePermission "${XL.HomeDir}\\GTC\\-", "read";

// Server needs read permissions on Nexaweb home directory
//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Read permissions on the "application-deployments" folder, the OIM deploy
// directory
permission java.io.FilePermission
"${oracle.home}\\j2ee\\home\\application-deployments\\Xellerate\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\home\\applications\\Xellerate\\-", "read,write,delete";

// OIM server invokes the java compiler. You need "execute"
// permissions on all files.
        permission java.io.FilePermission "<<ALL FILES>>", "execute";

// Socket permissions
// Basically you allow all permissions on nonprivileged sockets
// The multicast address should be the same as the one in
// xlconfig.xml for javagroups communication
        permission java.net.SocketPermission "*",
        "connect,listen,resolve,accept";
```

```
        permission java.net.SocketPermission "231.184.202.110",
        "connect,accept";

// Property permissions
// Read and write OIM properties
// Read XL.*, java.* and log4j.* properties
        permission java.util.PropertyPermission "XL.*", "read,write";
        permission java.util.PropertyPermission "*", "read, write";
        permission java.util.PropertyPermission "java.*", "read";
        permission java.util.PropertyPermission "log4j.", "read";
        permission java.util.PropertyPermission "user.dir", "read";

// Runtime permissions
// OIM server needs permissions to create its own class loader,
// get the class loader, modify threads and register shutdown
// hooks
        permission java.lang.RuntimePermission "createClassLoader";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "modifyThread";
        permission java.lang.RuntimePermission "modifyThreadGroup";
        permission java.lang.RuntimePermission "shutdownHooks";

// OIM server needs runtime permissions to generate and load
// classes in the below specified packages. Also access the
// declared members of a class.
permission java.lang.RuntimePermission
        "defineClassInPackage.com.thortech.xl.adapterGlue.ScheduleItemEvents";
permission java.lang.RuntimePermission
        "defineClassInPackage.com.thortech.xl.dataobj.rulegenerators";
permission java.lang.RuntimePermission
        "defineClassInPackage.com.thortech.xl.adapterGlue";
permission java.lang.RuntimePermission "accessDeclaredMembers";

// Reflection permissions
// Give permissions to access and invoke fields/methods from
// reflected classes.
        permission java.lang.reflect.ReflectPermission
                "suppressAccessChecks";

// Security permissions for OIM server
permission java.security.SecurityPermission "*";
permission javax.security.auth.AuthPermission "doAs";
permission javax.security.auth.AuthPermission "doPrivileged";
permission javax.security.auth.AuthPermission "getSubject";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission
                "createLoginContext";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission javax.security.auth.AuthPermission
                "getLoginConfiguration";
permission javax.security.auth.AuthPermission
                "setLoginConfiguration";

// SSL permission (for remote manager)
permission javax.net.ssl.SSLPermission  "getSSLSessionContext";
permission java.net.SocketPermission "*:1024-", "listen";
permission java.util.logging.LoggingPermission "control";
permission java.lang.RuntimePermission "enableContextClassLoaderOverride";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.io.SerializablePermission "enableSubstitution";
```

```
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission javax.security.auth.AuthPermission
               "getPolicy";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\Xellerate.err",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};

// Nexaweb server codebase permissions
grant codeBase "file:${oracle.home}/j2ee/home/applications/Nexaweb/-" {
// File permissions
permission java.io.FilePermission "${user.home}", "read, write";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\home\\application-deployments\\Nexaweb\\-",
"read,write,delete";

//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Property permissions
permission java.util.PropertyPermission "*", "read,write";

// Runtime permissions
// Nexaweb server needs permissions to create its own class loader,
// get the class loader etc.
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "setContextClassLoader";
permission java.lang.RuntimePermission  "setFactory";

// Nexaweb server security permissions to load the Cryptix
// extension
permission java.security.SecurityPermission "insertProvider.Cryptix";

// Socket permissions
// Permissions on all non-privileged ports.
permission java.net.SocketPermission "*:1024-", "listen, connect, resolve";

// Security permissions
permission javax.security.auth.AuthPermission "doAs";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission "createLoginContext";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.util.logging.LoggingPermission "control";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.io.SerializablePermission "enableSubstitution";
permission javax.security.auth.AuthPermission "getPolicy";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};
// The following are permissions given to codebase in the OIM server
```

```
// directory
grant codeBase "file:${XL.HomeDir}/-" {
// File permissions
permission java.io.FilePermission "${XL.HomeDir}\\config\\-", "read";
permission java.io.FilePermission "${XL.HomeDir}\\JavaTasks\\-", "read";
permission java.io.FilePermission "${XL.HomeDir}\\ScheduleTasks\\-", "read";
permission java.io.FilePermission "${XL.HomeDir}\\ThirdParty\\-", "read";
permission java.io.FilePermission "${XL.HomeDir}\\adapters\\-",
"read,write,delete";

//permission java.io.FilePermission "${nexaweb.home}\\-", "read";
// Socket permissions
permission java.net.SocketPermission "*", "listen";

// Property permissions
// Read XL.* and log4j.* properties
permission java.util.PropertyPermission "XL.*", "read";
permission java.util.PropertyPermission "log*", "read";

// Security permissions
permission javax.security.auth.AuthPermission "doAs";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission "createLoginContext";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.io.SerializablePermission "enableSubstitution";
permission java.util.logging.LoggingPermission "control";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.security.SecurityPermission "*";
permission javax.security.auth.AuthPermission "getLoginConfiguration";
permission javax.security.auth.AuthPermission "getPolicy";
permission javax.security.auth.AuthPermission "setLoginConfiguration";
permission java.security.SecurityPermission "insertProvider.Cryptix";

// Socket permissions
// Permissions on all nonprivileged ports.
permission java.net.SocketPermission "*:1024-","listen, connect, resolve";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};
```

Java 2 Security Permissions for Oracle Application Server Cluster

Note: The application might fail to start because of syntax errors in the policy files.

Be careful when editing the policy files. Oracle recommends that you use the policy tool provided by the JDK for editing the policy files. The tool is available in the following directory:

JAVA_HOME/jre/bin/policytool

To enable Java 2 Security for Oracle Identity Manager running on Oracle Application Server:

1. Modify the Oracle Application Server run configuration and add the `-Djava.security.manager` as a JVM option of the Oracle Application Server instance where Oracle Identity Manager is deployed. This change should be done in `$OC4J_HOME/opmn/conf/opmn.xml`.
2. Pass the following option to Oracle Application Server:

   ```
   -Djava.security.manager
   ```

   This option enables the Java 2 Security manager.
3. Check if the *`$ORACLEAS_HOME`*`/j2ee/<OC4J instance>/config/java2.policy` file exists. If it exists, edit it and add the Java 2 Security permissions listed in the "Policy File" section on page A-21.

   Note: If the `java2.policy` file does not exist, you have to create it.

Policy File

Perform the following in the `java2.policy` file:

Note:

- The instructions to change the code in the policy file are given in comments, which are in bold font.

- Make sure to change the Oracle Application Server instance name in the example below to reflect the Oracle Application Server on which you install Oracle Identity Manager. This example uses xlClusterMember for the instance name where Oracle Identity Manager is deployed.

- This `java2.policy` example is for Windows installation. For UNIX, ensure that you change `\\` between the directories name to `/` in every `permission java.io.FilePermission` property.

- Make sure to change the multicast IP `231.111.153.118` in this example to reflect the multicast IP address of the Oracle Identity Manager installation. You can find the Oracle Identity Manager multicast IP address in `xlconfig.xml`.

- You must update the path to the correct value for the location where GTC-RECON connector files are located. This example uses `C:\\file1\\file1` for the location of these files.

1. Find the following:

   ```
   grant { permission java.util.PropertyPermission
   "javax.xml.parsers.DocumentBuilderFactory" , "read";
   };
   ```

   Add the following to the preceding code:

   ```
   grant { permission java.util.PropertyPermission
   "javax.xml.parsers.DocumentBuilderFactory", "write";
   };
   ```

2. Find **/*Default Grants copied from the JDK default system policy*/** and add the following code to the grant:

```
//Added for OIM
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "*", "write";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.net.SocketPermission "*", "connect";
permission java.lang.RuntimePermission "accessClassInPackage.*";
permission javax.management.MBeanServerPermission "findMBeanServer";
permission javax.security.auth.AuthPermission "createLoginContext.*";

// For Nexaweb
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "setContextClassLoader";
permission java.util.PropertyPermission "nexaweb.logs", "read,write";
permission java.util.PropertyPermission
"sun.net.client.defaultConnectTimeout", "read,write";
permission java.util.PropertyPermission "sun.net.client.defaultReadTimeout",
"read,write";
permission java.lang.RuntimePermission "loadLibrary.*";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.net.SocketPermission    "*", "connect";
permission java.io.FilePermission       "<<ALL FILES>>", "read";
permission java.lang.RuntimePermission   "modifyThreadGroup";
permission oracle.oc4j.security.OC4JRuntimePermission "oracle.oc4j.OC4JOnly";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.jmx.server.mbeans.model.DefaultModelMBeanImpl#-", "*";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.jmx.server.mbeans.model.DefaultModelMBeanImpl#fireXMLConfigE
vent[default:j2eeType=OracleASJMSRouter]", "invoke";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.management.mbeans.JMSPersistence#-", "*";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.management.mbeans.JMSQueue#-", "*";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.management.mbeans.JMS#-", "*";
permission javax.management.MBeanPermission
"oracle.j2ee.ws.server.mgmt.runtime.mbean.ServerInterceptorGlobalRuntime#-","*"
;

//Change this to the original directory where logs are being geting created
//If logs are getting created in more then one directory ensure that you have
two entries for them here.
permission java.io.FilePermission "${oracle.home}\\opmn\\logs\\-",
"read,write,delete";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\xlClusterMember\\logs\\-", "read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\logs\\-",
"read,write,delete";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\xlClusterMember\\velocity.log", "read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\velocity.log",
"read,write,delete";
//This is added for the GTC-Recon Connector
/*
*  permission java.io.FilePermission "C:\\files\\file1\\-",
"read,write,delete";
*  property has been added for the path of directory where files are kept for
* GTC-RECON connector . Update the path to correct value prior to
* running the server.
```

```
*/
permission java.io.FilePermission "C:\\files\\file1\\-", "read,write,delete";

//Added for AQ
permission java.lang.RuntimePermission "accessDeclaredMembers";
```

3. In Custom Application Permissions, append the following code:

```
// Java code and extensions
// Trust java extensions
java.home}/lib/ext/-" {
permission java.security.AllPermission;
};

/*grant codeBase "file:${XL.HomeDir}/logs/-" {
permission java.security.AllPermission;
  };
*/

// Trust core java code
grant codeBase "file:${java.home}/lib/*" {
permission java.security.AllPermission;
};

// For java.home pointing to the JDK jre directory
grant codeBase "file:${java.home}/jre/lib/-" {
permission java.security.AllPermission;
};


// Grant All permissions to nexaweb commons jar file to be loaded from
grant codeBase
"file:${oracle.home}/j2ee/xlClusterMember/applib/nexaweb-common.jar" {
permission java.security.AllPermission;
};

// OIM codebase permissions
grant codeBase
"file:${oracle.home}/j2ee/xlClusterMember/applications/Xellerate/-" {

// File permissions
// Need read, write, and delete permissions on $OIM_HOME/config folder
// to read various config files, write the
// xlconfig.xml.{0,1,2..} files upon re-encryption and delete
// the last xlconfig.xml if the numbers go above 9.
        permission java.io.FilePermission "${XL.HomeDir}\\config\\-",
        "read, write, delete";
        permission java.io.FilePermission "${XL.HomeDir}\\-", "read";

        // Need read,write,delete permissions to generate adapter java
        // code, delete the .class file when the adapter is loaded into
        // the database
        permission java.io.FilePermission "${XL.HomeDir}\\adapters\\-",
 "read,write,delete";

        // This is required by the connectors and connector installer
        permission java.io.FilePermission
        "${XL.HomeDir}\\ConnectorDefaultDirectory\\-",
                "read,write,delete";
        permission java.io.FilePermission
                "${XL.HomeDir}\\adapters\\connectorResources\\-",
```

```
                "read,write,delete";

// Read Globalization resource bundle files for various
// locales
        permission java.io.FilePermission
        "${XL.HomeDir}\\adapters\\customResources\\-", "read";

// Read code from "JavaTasks", "ScheduleTask",
// "ThirdParty", "EventHandlers" folder
        permission java.io.FilePermission
        "${XL.HomeDir}\\EventHandlers\\-", "read";
        permission java.io.FilePermission
        "${XL.HomeDir}\\JavaTasks\\-", "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\ScheduleTask\\-", "read";
        permission java.io.FilePermission
        "${XL.HomeDir}\\ThirdParty\\-", "read";

// Required by the Generic Technology connector
        permission java.io.FilePermission "${XL.HomeDir}\\GTC\\-", "read";

// Server needs read permissions on Nexaweb home directory
//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Read permissions on the "applicatin-deployments" folder, the OIM deploy
// directory
permission java.io.FilePermission
"${oracle.home}\\j2ee\\xlClusterMember\\application-deployments\\Xellerate\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\xlClusterMember\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.io.FilePermission
        "${oracle.home}\\j2ee\\xlClusterMember\\applications\\Xellerate\\-",
"read,write,delete";

// OIM server invokes the java compiler. You need "execute"
// permissions on all files.
        permission java.io.FilePermission "<<ALL FILES>>", "execute";

// Socket permissions
// Basically we allow all permissions on nonprivileged sockets
// The multicast address should be the same as the one in
// xlconfig.xml for javagroups communication
        permission java.net.SocketPermission "*",
        "connect,listen,resolve,accept";
        permission java.net.SocketPermission "231.111.153.118",
        "connect,accept";

// Property permissions
// Read and write OIM properties
// Read XL.*, java.* and log4j.* properties
        permission java.util.PropertyPermission "XL.*", "read,write";
        permission java.util.PropertyPermission "*", "read, write";
        permission java.util.PropertyPermission "java.*", "read";
        permission java.util.PropertyPermission "log4j.", "read";
        permission java.util.PropertyPermission "user.dir", "read";

// Runtime permissions
```

```
// OIM server needs permissions to create its own class loader,
// get the class loader, modify threads and register shutdown
// hooks
        permission java.lang.RuntimePermission "createClassLoader";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "modifyThread";
        permission java.lang.RuntimePermission "modifyThreadGroup";
        permission java.lang.RuntimePermission "shutdownHooks";

// OIM server needs runtime permissions to generate and load
// classes in the below specified packages. Also access the
// declared members of a class.
        permission java.lang.RuntimePermission
        "defineClassInPackage.com.thortech.xl.adapterGlue.ScheduleItemEvents";
        permission java.lang.RuntimePermission
                "defineClassInPackage.com.thortech.xl.dataobj.rulegenerators";
        permission java.lang.RuntimePermission
                "defineClassInPackage.com.thortech.xl.adapterGlue";
        permission java.lang.RuntimePermission "accessDeclaredMembers";


// Reflection permissions
// Give permissions to access and invoke fields/methods from
// reflected classes.
        permission java.lang.reflect.ReflectPermission
                "suppressAccessChecks";

// Security permissions for OIM server
        permission java.security.SecurityPermission "*";
        permission javax.security.auth.AuthPermission "doAs";
        permission javax.security.auth.AuthPermission "doPrivileged";
        permission javax.security.auth.AuthPermission "getSubject";
        permission javax.security.auth.AuthPermission "modifyPrincipals";
        permission javax.security.auth.AuthPermission
                "createLoginContext";
        permission javax.security.auth.AuthPermission "createLoginContext.*";
        permission javax.security.auth.AuthPermission
                "getLoginConfiguration";
        permission javax.security.auth.AuthPermission
                "setLoginConfiguration";

// SSL permission (for remote manager)
       permission javax.net.ssl.SSLPermission  "getSSLSessionContext";
       permission java.net.SocketPermission "*:1024-", "listen";
       permission java.util.logging.LoggingPermission "control";
       permission java.lang.RuntimePermission
"enableContextClassLoaderOverride";
       permission java.io.SerializablePermission
"enableSubclassImplementation";
       permission java.io.SerializablePermission "enableSubstitution";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission javax.security.auth.AuthPermission
                "getPolicy";
permission java.util.PropertyPermission "javax.*", "read,write";
permission oracle.security.jazn.JAZNPermission "getRealmManager";
};
```

```
// Nexaweb server codebase permissions
grant codeBase
"file:${oracle.home}/j2ee/xlClusterMember/applications/Nexaweb/-" {
// File permissions
        permission java.io.FilePermission "${user.home}", "read, write";
        permission java.io.FilePermission
"${oracle.home}\\j2ee\\xlClusterMember\\application-deployments\\Nexaweb\\-",
"read,write,delete";
//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Property permissions
permission java.util.PropertyPermission "*", "read,write";

// Runtime permissions
// Nexaweb server needs permissions to create its own class loader,
// get the class loader etc.
        permission java.lang.RuntimePermission "createClassLoader";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "setContextClassLoader";
        permission java.lang.RuntimePermission  "setFactory";

// Nexaweb server security permissions to load the Cryptix
// extension
        permission java.security.SecurityPermission
        "insertProvider.Cryptix";

// Socket permissions
// Permissions on all non-privileged ports.
        permission java.net.SocketPermission "*:1024-",
                "listen, connect, resolve";

// Security permissions
        permission javax.security.auth.AuthPermission "doAs";
        permission javax.security.auth.AuthPermission "modifyPrincipals";
        permission javax.security.auth.AuthPermission "createLoginContext";
        permission javax.security.auth.AuthPermission "createLoginContext.*";
        permission java.util.logging.LoggingPermission "control";
        permission java.io.SerializablePermission
"enableSubclassImplementation";
        permission java.io.SerializablePermission "enableSubstitution";
        permission javax.security.auth.AuthPermission
        "getPolicy";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\xlClusterMember\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};

// The following are permissions given to codebase in the OIM server
// directory
grant codeBase "file:${XL.HomeDir}/-" {
// File permissions
        permission java.io.FilePermission "${XL.HomeDir}\\config\\-",
                "read";
```

```
        permission java.io.FilePermission "${XL.HomeDir}\\JavaTasks\\-",
                "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\ScheduleTasks\\-", "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\ThirdParty\\-", "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\adapters\\-", "read,write,delete";

//permission java.io.FilePermission "${nexaweb.home}\\-", "read";
// Socket permissions
        permission java.net.SocketPermission "*", "listen";

// Property permissions
// Read XL.* and log4j.* properties
        permission java.util.PropertyPermission "XL.*", "read";
        permission java.util.PropertyPermission "log*", "read";

// Security permissions
        permission javax.security.auth.AuthPermission "doAs";
        permission javax.security.auth.AuthPermission "modifyPrincipals";
        permission javax.security.auth.AuthPermission "createLoginContext";
permission java.io.SerializablePermission "enableSubclassImplementation";
       permission java.io.SerializablePermission "enableSubstitution";
permission java.util.logging.LoggingPermission "control";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.security.SecurityPermission "*";
permission javax.security.auth.AuthPermission
        "getLoginConfiguration";
permission javax.security.auth.AuthPermission
"getPolicy";
        permission javax.security.auth.AuthPermission
                "setLoginConfiguration";
permission java.security.SecurityPermission
"insertProvider.Cryptix";

// Socket permissions
// Permissions on all non-privileged ports.
permission java.net.SocketPermission "*:1024-","listen, connect, resolve";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\xlClusterMember\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};
```

Policy File

The following is the sample `java2.policy` file after Oracle Identity Manager policy has been added:

```
/*
 * Standard policy file for Oracle Application Server
 *
 *    When this file is in use the System property ${oracle.home} must
 *    be set to $ORACLE_HOME or to the value of $ORACLE_HOME.
 *
 *    When this file is in use via OPMN the System property
```

```
 *    ${oracle.oc4j.instancename}
 *    is used to identify the instance-level connector jars.
 *
 *    This file grants AllPermission to "oc4j code"
 *    oc4j code is code used either directly or indirectly by the app server
 *    itself. Including code generated for ejb wrappers.
 *    See oc4j.jar!boot.xml for a complete list. Currently this file
 *    only lists jars that need permissions. Others can be
 *    added if neccessary.
 *
 *    In a future release the grants will be refined so that
 *    only the Permissions actually needed by Oracle Application Server
 *    code will be granted.
 *
 *    Calls to accessController.doPrivileged have been added to Oracle
 *    Application Server with the intention that the application code only
 *    be granted the Permissions needed by actions it performs directly.
 *    It should not be granted Permissions required by J2EE
 *    operations.
 *
 *    For example if a Servlet (or jsp) forwards to a .jsp it does not
 *    need Permission to read and compile the .jsp.  Similarly the
 *    application code associated with an ejb that specifies container
 *    managed persistence does not need Permission to create a socket
 *    talking to the database holding the underlying data. But an EJB
 *    using bean managed persistence does need such Permission.
 */
grant codebase "file:${oracle.home}/j2ee/home/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/lib/*" {
    permission java.security.AllPermission;
};


grant codebase "file:${oracle.home}/jlib/-" {
    permission java.security.AllPermission;
};


grant codebase "file:${oracle.home}/bc4j/jlib/*" {
    permission java.security.AllPermission;
};

grant codeBase "file:${oracle.home}/toplink/jlib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/dms/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/diagnostics/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/jdbc/lib/ojdbc14dms.jar" {
    permission java.security.AllPermission;
};
```

```
grant codebase "file:${oracle.home}/dbjava/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/sqlj/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/javacache/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/uddi/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/xdk/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/opmn/lib/*" {
    permission java.security.AllPermission;
};


grant codebase "file:${oracle.home}/webservices/lib/*" {
    permission java.security.AllPermission;
};



grant codeBase "file:${oracle.home}/javavm/lib/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/jsp/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/lib/*" {
    permission java.security.AllPermission;
};


/** EJB skeleton/tie & BCEL proxy support **/

grant codeBase "file:generated/by/proxy" {
    permission java.security.AllPermission;
};

grant codeBase
"file://generated/by/oracle.j2ee.connector.proxy.BCELProxyClassLoader" {
    permission java.security.AllPermission;
};

/**
* Miscellaneous grants to jars distributed as part of oc4j that can be used
* in various ways
*/
```

```
grant codebase
"file:${oracle.home}/j2ee/home/connectors/OracleASjms/OracleASjms/gjra.jar" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/connectors/OracleASjms/Oracl
eASjms/gjra.jar" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/home/connectors/datasources/datasources/datasources.jar"
{
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/connectors/datasources/datas
ources/datasources.jar" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/jsp/lib/*" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/jsp/lib/taglib/ojsputil.jar" {
    permission java.security.AllPermission;
};

/* GRANTS TO DEFAULT APPLICATIONS */

grant codebase
"file:${oracle.home}/j2ee/xlClusterMember/application-deployments/ascontrol/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/asco
ntrol/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/xlClusterMember/applications/ascontrol/-"
{
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/ascontrol/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/xlClusterMember/application-deployments/default/-" {
    permission java.security.AllPermission;
};
```

```
grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/defa
ult/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/xlClusterMember/applications/default/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/default/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/application-deployments/javasso/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/java
sso/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/javasso/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/javasso/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/application-deployments/usermbean/-"
{
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/user
mbean/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/usermbean/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/usermbean/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/xlClusterMember/application-deployments/admin_ejb/-" {
    permission java.security.AllPermission;
};

grant codebase
```

```
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/admi
n_ejb/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/admin_ejb.jar" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/admin_ejb/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/applications/admin_ejb/-" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/jmsrouter-ejb.jar" {
    permission java.security.AllPermission;
};

grant codebase "file:${oracle.home}/j2ee/home/applications/jmsrouter" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/xlClusterMember/application-deployments/JMXSoapAdapter-w
eb/-" {
    permission java.security.AllPermission;
};

grant codebase
"file:${oracle.home}/j2ee/${oracle.oc4j.instancename}/application-deployments/JMXS
oapAdapter-web/-" {
    permission java.security.AllPermission;
};


grant { permission java.util.PropertyPermission "j2ee.home", "read"; } ;
grant { permission java.util.PropertyPermission "java.home", "read"; } ;
grant { permission java.util.PropertyPermission "javax.xml.soap.SOAPFactory",
"read"; } ;
grant { permission java.util.PropertyPermission "javax.activation.debug" , "read";
} ;
grant { permission java.util.PropertyPermission
"javax.xml.parsers.DocumentBuilderFactory" , "read"; } ;
grant { permission java.util.PropertyPermission
"javax.xml.parsers.DocumentBuilderFactory", "write"; };
grant { permission java.util.PropertyPermission "javax.xml.soap.MessageFactory" ,
"read"; } ;
grant { permission java.util.PropertyPermission "jdbc.nontx.autocommit" , "read";
} ;
grant { permission java.util.PropertyPermission "mail.URLName.dontencode" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oc4j.jmx.event.interval" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oc4j.jmx.heartbeat.interval" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.defaultNChar" ,
```

```
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.DMSStatementMetrics"
, "read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.J2EE13Compliant" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.TcpNoDelay" , "read";
} ;
grant { permission java.util.PropertyPermission
"oracle.jdbc.useFetchSizeWithLongColumn" , "read"; } ;
grant { permission java.util.PropertyPermission "oracle.jdbc.V8Compatible" ,
"read"; } ;
grant { permission java.util.PropertyPermission "oracle.jserver.version" , "read";
} ;
grant { permission java.util.PropertyPermission "oracle.xml.parser.debugmode" ,
"read"; } ;
grant { permission java.util.PropertyPermission
"oracle.xml.parser.default.character.set" , "read"; } ;
grant { permission java.util.PropertyPermission "oracle.xml.xslt.jdwp", "read"; };
grant { permission java.util.PropertyPermission "orasaaj.soapversion" , "read"; }
;
grant { permission java.util.PropertyPermission "org.apache.commons.logging.Log" ,
"read"; } ;
grant { permission java.util.PropertyPermission
"org.apache.commons.logging.LogFactory" , "read"; } ;
grant { permission java.util.PropertyPermission "PersistenceManagerDebug" ,
"read"; } ;
grant { permission java.util.PropertyPermission "pro.debug" , "read"; } ;
grant { permission java.util.PropertyPermission "sqlj.runtime" , "read"; } ;
grant { permission java.util.PropertyPermission "transaction.debug" , "read"; } ;
grant { permission java.util.PropertyPermission "user.home" , "read"; } ;
grant { permission java.util.PropertyPermission "user.name" , "read"; } ;
grant { permission java.util.PropertyPermission "rmi.verbose" , "read"; } ;
grant { permission java.util.PropertyPermission "AssociateUserToThread", "read";
};
grant { permission java.util.PropertyPermission
"toplink.cts.collection.checkParameters", "read"; };
grant { permission java.util.PropertyPermission "AllowZeroInPK", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.Modules", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.Nagle", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.cookies.hosts.accept",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.cookies.hosts.reject",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.cookies.save", "read";
};
grant { permission java.util.PropertyPermission "HTTPClient.deferStreamed",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.disableKeepAlives",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.disable_pipelining",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.dontChunkRequests",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.dontTimeoutRespBody",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.forceHTTP_1.0",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.log.level", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.nonProxyHosts",
"read"; };
```

```
grant { permission java.util.PropertyPermission "HTTPClient.socket.idleTimeout",
"read"; };
grant { permission java.util.PropertyPermission "HTTPClient.socksHost", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.socksPort", "read"; };
grant { permission java.util.PropertyPermission "HTTPClient.socksVersion", "read";
};
grant { permission java.util.PropertyPermission "JavaClass.debug", "read"; };
grant { permission java.util.PropertyPermission "LoadBalanceOnLookup", "read"; };
grant { permission java.util.PropertyPermission "SQLLog", "read"; };
grant { permission java.util.PropertyPermission "USE_JAAS", "read"; };
grant { permission java.util.PropertyPermission "com.sun.enterprise.home", "read";
};
grant { permission java.util.PropertyPermission
"customFinderMethod.noLazyLoading", "read"; };
grant { permission java.util.PropertyPermission "debug", "read"; };
grant { permission java.util.PropertyPermission "default.cmp.pm", "read"; };
grant { permission java.util.PropertyPermission "ejb.debug.verbose", "read"; };
grant { permission java.util.PropertyPermission "findByPrimaryKey.noLazyLoading",
"read"; };
grant { permission java.util.PropertyPermission "http.nonProxyHosts", "read"; };
grant { permission java.util.PropertyPermission "http.proxyHost", "read"; };
grant { permission java.util.PropertyPermission "http.proxyPort", "read"; };
grant { permission java.util.PropertyPermission "java.ext.dirs", "read"; };
grant { permission java.util.PropertyPermission "java.class.path", "read"; };
grant { permission java.util.PropertyPermission
"javax.xml.parsers.SAXParserFactory", "read"; };
grant { permission java.util.PropertyPermission "jca.connection.debug", "read"; };
grant { permission java.util.PropertyPermission "log4j.configDebug", "read"; };
grant { permission java.util.PropertyPermission "log4j.configuration", "read"; };
grant { permission java.util.PropertyPermission "log4j.debug", "read"; };
grant { permission java.util.PropertyPermission "log4j.defaultInitOverride",
"read"; };
grant { permission java.util.PropertyPermission "log4j.disable", "read"; };
grant { permission java.util.PropertyPermission "log4j.disableOverride", "read";
};
grant { permission java.util.PropertyPermission "oneToOneJoin", "read"; };
grant { permission java.util.PropertyPermission "sun.boot.class.path", "read"; };
grant { permission java.util.PropertyPermission "toplink.changePolicy", "read"; };
grant { permission java.util.PropertyPermission
"toplink.cts.collection.checkParameters", "read"; };
grant { permission java.util.PropertyPermission
"toplink.cts.collection.checkTransaction", "read"; };
grant { permission java.util.PropertyPermission
"toplink.defaultmapping.dbTableGenSetting", "read"; };
grant { permission java.util.PropertyPermission
"toplink.defaultmapping.useExtendedTableNames", "read"; };
grant { permission java.util.PropertyPermission "toplink.log.destination", "read";
};
grant { permission java.util.PropertyPermission "toplink.log.level", "read"; };
grant { permission java.util.PropertyPermission "toplink.xml.platform", "read"; };
grant { permission java.util.PropertyPermission "upload.buflen", "read"; };
grant { permission java.util.PropertyPermission "user.dir", "read"; };
grant { permission java.util.PropertyPermission
"javax.xml.soap.SOAPConnectionFactory", "read";};
grant { permission java.util.PropertyPermission "HTTPClient.socket.idleTimeout",
"write";};


/* JDK  */
```

```
grant codebase "file:${java.home}/../lib/tools.jar" {
    permission java.security.AllPermission;
};

grant codeBase "file:${java.home}/lib/ext/*" {
permission java.security.AllPermission;
};

/* Default Grants copied from the JDK default system policy. */

grant {
// "standard" properties that can be read by anyone.

permission java.util.PropertyPermission "java.version", "read";
permission java.util.PropertyPermission "java.vendor", "read";
permission java.util.PropertyPermission "java.vendor.url", "read";
permission java.util.PropertyPermission "java.class.version", "read";
permission java.util.PropertyPermission "os.name", "read";
permission java.util.PropertyPermission "os.version", "read";
permission java.util.PropertyPermission "os.arch", "read";
permission java.util.PropertyPermission "file.separator", "read";
permission java.util.PropertyPermission "path.separator", "read";
permission java.util.PropertyPermission "line.separator", "read";

permission java.util.PropertyPermission "java.specification.version", "read";
permission java.util.PropertyPermission "java.specification.vendor", "read";
permission java.util.PropertyPermission "java.specification.name", "read";

permission java.util.PropertyPermission "java.vm.specification.version", "read";
permission java.util.PropertyPermission "java.vm.specification.vendor", "read";
permission java.util.PropertyPermission "java.vm.specification.name", "read";
permission java.util.PropertyPermission "java.vm.version", "read";
permission java.util.PropertyPermission "java.vm.vendor", "read";
permission java.util.PropertyPermission "java.vm.name", "read";


/* The following are granted by the default jdk policy but are considered
* unsafe and are omitted by this policy file */

//permission java.lang.RuntimePermission "stopThread";
//permission java.net.SocketPermission "localhost:1024-", "listen";

permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "*", "write";
permission java.lang.RuntimePermission "queuePrintJob";
permission java.net.SocketPermission "*", "connect";
permission java.lang.RuntimePermission "accessClassInPackage.*";
permission javax.management.MBeanServerPermission "findMBeanServer";
permission javax.security.auth.AuthPermission "createLoginContext.*";

// For Nexaweb
permission java.lang.RuntimePermission "getClassLoader";
permission java.lang.RuntimePermission "setContextClassLoader";
permission java.util.PropertyPermission "nexaweb.logs", "read,write";
permission java.util.PropertyPermission
"sun.net.client.defaultConnectTimeout", "read,write";
permission java.util.PropertyPermission "sun.net.client.defaultReadTimeout",
"read,write";
permission java.lang.RuntimePermission "loadLibrary.*";
permission java.lang.RuntimePermission "queuePrintJob";
```

```
permission java.net.SocketPermission    "*", "connect";
permission java.io.FilePermission       "<<ALL FILES>>", "read";
permission java.lang.RuntimePermission   "modifyThreadGroup";
permission oracle.oc4j.security.OC4JRuntimePermission "oracle.oc4j.OC4JOnly";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.jmx.server.mbeans.model.DefaultModelMBeanImpl#-", "*";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.jmx.server.mbeans.model.DefaultModelMBeanImpl#fireXMLConfigEven
t[default:j2eeType=OracleASJMSRouter]", "invoke";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.management.mbeans.JMSPersistence#-", "*";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.management.mbeans.JMSQueue#-", "*";
permission javax.management.MBeanPermission
"oracle.oc4j.admin.management.mbeans.JMS#-", "*";
permission javax.management.MBeanPermission
"oracle.j2ee.ws.server.mgmt.runtime.mbean.ServerInterceptorGlobalRuntime#-","*";

//Change this to the original directory where logs are being geting created
//If logs are getting created in more then one directory ensure that you have two
entries for them here.
permission java.io.FilePermission "${oracle.home}\\opmn\\logs\\-",
"read,write,delete";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\xlClusterMember\\logs\\-", "read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\logs\\-",
"read,write,delete";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\xlClusterMember\\velocity.log", "read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\velocity.log",
"read,write,delete";
//This is added for the GTC-Recon Connector
permission java.io.FilePermission "C:\\files\\file1\\-", "read,write,delete";


};

/**
** Add Custom Application Permission Grants Below
**/
// Java code and extensions
// Trust java extensions
grant codeBase "file:${java.home}/lib/ext/-" {
permission java.security.AllPermission;
};

/*grant codeBase "file:${XL.HomeDir}/logs/-" {
permission java.security.AllPermission;
  };
*/

// Trust core java code
grant codeBase "file:${java.home}/lib/*" {
permission java.security.AllPermission;
};

// For java.home pointing to the JDK jre directory
grant codeBase "file:${java.home}/jre/lib/-" {
permission java.security.AllPermission;
};
```

```
// Grant All permissions to nexaweb commons jar file to be loaded from
grant codeBase
"file:${oracle.home}/j2ee/xlClusterMember/applib/nexaweb-common.jar" {
permission java.security.AllPermission;
};

// OIM codebase permissions
grant codeBase "file:${oracle.home}/j2ee/xlClusterMember/applications/Xellerate/-"
{

// File permissions
// Need read,write,delete permissions on $OIM_HOME/config folder
// to read various config files, write the
// xlconfig.xml.{0,1,2..} files upon re-encryption and delete
// the last xlconfig.xml if the numbers go above 9.
        permission java.io.FilePermission "${XL.HomeDir}\\config\\-",
        "read, write, delete";
        permission java.io.FilePermission "${XL.HomeDir}\\-", "read";

// Need read,write,delete permissions to generate adapter java
// code, delete the .class file when the adapter is loaded into
// the database
        permission java.io.FilePermission "${XL.HomeDir}\\adapters\\-",
"read,write,delete";

// This is required by the connectors and connector installer
        permission java.io.FilePermission
        "${XL.HomeDir}\\ConnectorDefaultDirectory\\-",
                "read,write,delete";
        permission java.io.FilePermission
                "${XL.HomeDir}\\adapters\\connectorResources\\-",
                "read,write,delete";

// Read Globalization resource bundle files for various
// locales
        permission java.io.FilePermission
        "${XL.HomeDir}\\adapters\\customResources\\-", "read";

// Read code from "JavaTasks", "ScheduleTask",
// "ThirdParty", "EventHandlers" folder
        permission java.io.FilePermission
        "${XL.HomeDir}\\EventHandlers\\-", "read";
        permission java.io.FilePermission
        "${XL.HomeDir}\\JavaTasks\\-", "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\ScheduleTask\\-", "read";
        permission java.io.FilePermission
        "${XL.HomeDir}\\ThirdParty\\-", "read";

// Required by the Generic Technology connector
        permission java.io.FilePermission "${XL.HomeDir}\\GTC\\-", "read";

// Server needs read permissions on Nexaweb home directory
//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Read permissions on the "applicatin-deployments" folder, the OIM deploy
// directory
permission java.io.FilePermission
```

```
"${oracle.home}\\j2ee\\xlClusterMember\\application-deployments\\Xellerate\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\xlClusterMember\\-",
"read,write,delete";
permission java.io.FilePermission "${oracle.home}\\j2ee\\home\\-",
"read,write,delete";
permission java.io.FilePermission
"${oracle.home}\\j2ee\\xlClusterMember\\applications\\Xellerate\\-",
"read,write,delete";

// OIM server invokes the java compiler. You need "execute"
// permissions on all files.
        permission java.io.FilePermission "<<ALL FILES>>", "execute";

// Socket permissions
// Basically you allow all permissions on nonprivileged sockets
// The multicast address should be the same as the one in
// xlconfig.xml for javagroups communication
        permission java.net.SocketPermission "*",
        "connect,listen,resolve,accept";
        permission java.net.SocketPermission "231.111.153.118",
        "connect,accept";

// Property permissions
// Read and write OIM properties
// Read XL.*, java.* and log4j.* properties
        permission java.util.PropertyPermission "XL.*", "read,write";
        permission java.util.PropertyPermission "*", "read, write";
        permission java.util.PropertyPermission "java.*", "read";
        permission java.util.PropertyPermission "log4j.", "read";
        permission java.util.PropertyPermission "user.dir", "read";

// Runtime permissions
// OIM server needs permissions to create its own class loader,
// get the class loader, modify threads and register shutdown
// hooks
        permission java.lang.RuntimePermission "createClassLoader";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "modifyThread";
        permission java.lang.RuntimePermission "modifyThreadGroup";
        permission java.lang.RuntimePermission "shutdownHooks";

// OIM server needs runtime permissions to generate and load
// classes in the below specified packages. Also access the
// declared members of a class.
        permission java.lang.RuntimePermission
        "defineClassInPackage.com.thortech.xl.adapterGlue.ScheduleItemEvents";
        permission java.lang.RuntimePermission
                "defineClassInPackage.com.thortech.xl.dataobj.rulegenerators";
        permission java.lang.RuntimePermission
                "defineClassInPackage.com.thortech.xl.adapterGlue";
        permission java.lang.RuntimePermission "accessDeclaredMembers";


// Reflection permissions
// Give permissions to access and invoke fields/methods from
// reflected classes.
        permission java.lang.reflect.ReflectPermission
                "suppressAccessChecks";
```

```
// Security permissions for OIM server
        permission java.security.SecurityPermission "*";
        permission javax.security.auth.AuthPermission "doAs";
        permission javax.security.auth.AuthPermission "doPrivileged";
        permission javax.security.auth.AuthPermission "getSubject";
        permission javax.security.auth.AuthPermission "modifyPrincipals";
        permission javax.security.auth.AuthPermission
                "createLoginContext";
        permission javax.security.auth.AuthPermission "createLoginContext.*";
        permission javax.security.auth.AuthPermission
                "getLoginConfiguration";
        permission javax.security.auth.AuthPermission
                "setLoginConfiguration";

// SSL permission (for remote manager)
        permission javax.net.ssl.SSLPermission  "getSSLSessionContext";
permission java.net.SocketPermission "*:1024-", "listen";
permission java.util.logging.LoggingPermission "control";
permission java.lang.RuntimePermission "enableContextClassLoaderOverride";
permission java.io.SerializablePermission "enableSubclassImplementation";
        permission java.io.SerializablePermission "enableSubstitution";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
        permission javax.security.auth.AuthPermission
                "getPolicy";
permission java.util.PropertyPermission "javax.*", "read,write";
permission oracle.security.jazn.JAZNPermission "getRealmManager";
};


// Nexaweb server codebase permissions
grant codeBase "file:${oracle.home}/j2ee/xlClusterMember/applications/Nexaweb/-" {
// File permissions
        permission java.io.FilePermission "${user.home}", "read, write";
        permission java.io.FilePermission

"${oracle.home}\\j2ee\\xlClusterMember\\application-deployments\\Nexaweb\\-",
"read,write,delete";
//permission java.io.FilePermission "${nexaweb.home}\\-", "read";

// Property permissions
permission java.util.PropertyPermission "*", "read,write";

// Runtime permissions
// Nexaweb server needs permissions to create its own class loader,
// get the class loader etc.
        permission java.lang.RuntimePermission "createClassLoader";
        permission java.lang.RuntimePermission "getClassLoader";
        permission java.lang.RuntimePermission "setContextClassLoader";
        permission java.lang.RuntimePermission  "setFactory";

// Nexaweb server security permissions to load the Cryptix
// extension
        permission java.security.SecurityPermission
        "insertProvider.Cryptix";

// Socket permissions
```

```
// Permissions on all non-privileged ports.
        permission java.net.SocketPermission "*:1024-",
                "listen, connect, resolve";

// Security permissions
permission javax.security.auth.AuthPermission "doAs";
permission javax.security.auth.AuthPermission "modifyPrincipals";
permission javax.security.auth.AuthPermission "createLoginContext";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.util.logging.LoggingPermission "control";
permission java.io.SerializablePermission "enableSubclassImplementation";
permission java.io.SerializablePermission "enableSubstitution";
permission javax.security.auth.AuthPermission "getPolicy";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\xlClusterMember\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};

// The following are permissions given to codebase in the OIM server
// directory
grant codeBase "file:${XL.HomeDir}/-" {
// File permissions
        permission java.io.FilePermission "${XL.HomeDir}\\config\\-",
                "read";
        permission java.io.FilePermission "${XL.HomeDir}\\JavaTasks\\-",
                "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\ScheduleTasks\\-", "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\ThirdParty\\-", "read";
        permission java.io.FilePermission
                "${XL.HomeDir}\\adapters\\-", "read,write,delete";

//permission java.io.FilePermission "${nexaweb.home}\\-", "read";
// Socket permissions
        permission java.net.SocketPermission "*", "listen";

// Property permissions
// Read XL.* and log4j.* properties
        permission java.util.PropertyPermission "XL.*", "read";
        permission java.util.PropertyPermission "log*", "read";

// Security permissions
        permission javax.security.auth.AuthPermission "doAs";
        permission javax.security.auth.AuthPermission "modifyPrincipals";
        permission javax.security.auth.AuthPermission "createLoginContext";
permission java.io.SerializablePermission "enableSubclassImplementation";
        permission java.io.SerializablePermission "enableSubstitution";
permission java.util.logging.LoggingPermission "control";
permission javax.security.auth.AuthPermission "createLoginContext.*";
permission java.security.SecurityPermission "*";
        permission javax.security.auth.AuthPermission
        "getLoginConfiguration";
        permission javax.security.auth.AuthPermission
                "getPolicy";
```

```
        permission javax.security.auth.AuthPermission
                "setLoginConfiguration";
permission java.security.SecurityPermission
"insertProvider.Cryptix";

// Socket permissions
// Permissions on all non-privileged ports.
permission java.net.SocketPermission "*:1024-","listen, connect, resolve";
permission java.net.SocketPermission "*:*", "connect,resolve";
permission java.lang.RuntimePermission "createClassLoader";
permission java.lang.RuntimePermission "getClassLoader";
permission java.util.PropertyPermission "*", "read";
permission java.util.PropertyPermission "LoadBalanceOnLookup", "read,write";
permission java.io.FilePermission "${oracle.home}\\j2ee\\xlClusterMember\\-",
"read,write,delete";
permission java.util.PropertyPermission "javax.*", "read,write";
};
```

Index

O

P

R

S

T

X