Oracle® Identity Manager

Best Practices Guide

Release 9.0.3

B32451-01

February 2007

ORACLE®

Oracle Identity Manager Best Practices Guide, Release 9.0.3

B32451-01

Primary Author: Don Gosselin

Contents

Preface

This preface introduces you to the *Oracle Identity Manager Best Practices Guide* discussing the intended audience and conventions of this document. It also includes a list of related Oracle documents.

> **Note:** This is a transitional release following Oracle's acquisition of Thor Technologies. Some parts of the product and documentation still refer to the original Thor company name and Xellerate product name and will be rebranded in future releases.

Audience

The *Oracle Identity Manager Best Practices Guide* is intended for Database Administrators, System Administrators, and developers who plan to use Oracle Identity Manager extensively in production environments.

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, seven days a week. For TTY support, call 800.446.2398.

Related Documents

This guide assumes that you have read and understood the following documents:

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

- *Oracle Identity Manager Installation Guide for JBoss*
- *Oracle Identity Manager Installation Guide for WebLogic*
- *Oracle Identity Manager Installation Guide for WebSphere*
- *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 Guide*
- *Oracle Identity Manager Audit Report Developer Guide*
- *Oracle Identity Manager API Usage Guide*
- *Oracle Identity Manager Glossary of Terms*

Documentation Updates

Oracle is committed to delivering the best and most recent information available. For information about updates to the Oracle Identity Manager 9.0 documentation set, visit Oracle Technology Network at

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

Conventions

The following text conventions are used in this document:

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

1

Using the Deployment Manager

The Deployment Manager enables developers to move an Oracle Identity Manager deployment from one server to another while minimizing the problems that often occur during a migration. The Deployment Manager allows multiple developers to work on different parts of a deployment and upload only the part of the deployment that they have changed, rather than waiting for the entire deployment to be rebuilt.

> **Caution:** Be aware that the most recently imported data overwrites the previous data. Developer should not export data that can overwrite the work of another developer.

This chapter discusses the following topics:

Features of the Deployment Manager

The Deployment Manager helps migrate Oracle Identity Manager deployments from one server environment to another, such as from a test environment to a staging environment, or from a staging environment to a production environment.

The Deployment Manager provides the following benefits:

- Updating individual components of a deployment in different test environments
- Identifying objects associated with components to be exported, so that those resources can be included
- Providing information on exported files
- Allowing the exporter to add comments

The Deployment Manager handles the following types of information:

- Resource objects
- Adapters
- IT resource types
- User-defined forms
- Organizations
- User-defined field definitions
- Rule definitions
- Email definitions
- System and error Codes
- Lookup definitions
- User groups
- Password policies
- Access policies
- Scheduled tasks

Limitations of the Deployment Manager

The following are important limitations of the Deployment Manager:

- **Merge Utility**: The Deployment Manager is not a merge utility.

  It cannot handle changes done in both production and test environments. It will replace the object in target system with what is in the XML.

- **Version Control Utility**: Deployment Manager does not track versions of imported files, and does not provide rollback functionality.

  You can only use it as a means to move data between environments.

- **Code Moving**: Deployment manager does not move `.jar` files in JavaTasks or other locations.

  You must do this manually.

- **Custom Labels Move**: Deployment Manager does not move labels defined in the customResources.properties file or the property files in the connectorResources directory. You must do this manually.

Export System Objects Only when Necessary

Only export or import system objects, for example, Request, Xellerate User, and System Administrator, when it is absolutely necessary. Exporting them from testing and staging environments into production can cause problems. If possible, exclude system objects when exporting or importing data.

Some circumstances might require you to export or import system objects, for example, when defining Trusted Source Reconciliation on Xellerate Users resource objects.

Export Related Groups of Objects

Oracle recommends that you use the Deployment Manager to export sets of related objects. A unit of export should be a collection of logical items that you want to group together.

Avoid exporting everything in the database in one operation, or exporting items one at a time. For example, suppose that you manage an integration between Oracle Identity Manager and a target system that includes processes, resource objects, adapters, IT resource type definitions, IT resource definitions, scheduled tasks, and so on. For this environment, you should create groups of related objects before exporting.

For example, if you use the same email definitions in multiple integrations, you should export the email definitions as one unit, and integrations as different unit. This enables you to import changes to email definitions independently of target system integration changes. Or, if multiple resources use the same IT resource type definition, you can export and import the type definition separately from other data.

You can import one or more sets of exported data at a time. For example, you can import a resource object definition, email definition, and IT resource type definition in one operation.

Group Definition Data and Operational Data Separately

You should group and export definition data and operational data separately.

You configure definition data in test and staging. Definition includes resource objects, processes, rules, and so on.

You typically configure operational data in production. Operational data includes groups and group permissions. The test and staging servers normally do not include this data.

By grouping data according to where it is changed, you know what data goes to test and staging, and what goes to production. For example, if approval processes are changed in production, you should group approval processes and export them with other operational data.

Use Logical Naming Conventions for Versions of a Form

You often revise forms multiple times before exporting them. Avoid generic names, for example, "v23" to differentiate among versions of a form. Create meaningful names, for example, "Before Pre Production" or "After Production Verification". Do not use special characters, including double quotes, in version names.

Caution: The Deployment Manager keeps track of imported components and structures, but not of completed imports. After an import is completed, you cannot roll it back to a previous version. A new import is required.

Export Root to Preserve a Complete Organizational Hierarchy

When you export a leaf or an organization in an organizational hierarchy, only one dependency level is exported. To export a complete organizational hierarchy you must export the root of the hierarchy.

Provide Clear Export Descriptions

The Deployment Manager records some information automatically, for example, the date of the export, who performed the export, and the source database. You should also provide a meaningful description of the content of the export, for example, "resource definition after xxx attributes added in reconciliation". This informs the importer of the file of the contents of the data being imported.

Check All Warnings Before Importing

When importing to production, check all the warnings before completing the import. Treat each warning seriously.

Check Dependencies Before Exporting Data

The wizard in the top right pane shows resources that must be available in the target system.

The following types of dependencies may occur:

1. If the resources are already available in the target system, they do not need to be exported.
2. If the resources are new (not in the target system), they must be exported.
3. If the target system may or may not include the resources, such as lookups, IT resource definitions, or others that are reused, then record the data and export it in a separate file so it can be imported if necessary.

Matching Scheduled Task Parameters

Scheduled tasks depend on certain parameters to run properly. You can import scheduled task parameters to the production server. Table 1–1 shows the rules for determining how to import scheduled tasks. Note that parameters may be available for tasks that no longer reside on the target system.

Table 1–1 Parameter Import Rules

Parameter Exists in Target System	Parameter Exists in the XML File	Action Taken
Yes	No	Remove the parameter from the target system.
No	Yes	Add the parameter and current value from the XML file.
Yes	Yes	Use the more recent value of the parameter.

Compile Adapters and Enable Scheduled Tasks

After an import operation, adapters are set to recompile and scheduled tasks are disabled. This prevents these items from running prior to configuring their associated resources and settings.

After importing the classes and adjusting the task attributes, manually recompile adapters and enable scheduled tasks.

Export Entity Adapters Separately

Entity adapters are modified to bring just the entity adapter, not their usage. You must separately export each use of an entity adapter with a data object by exporting the data object. Exporting a data object exports all the adapters and event handlers attached to the object along with the permissions on the object. You need to pay special attention when exporting data objects. For example, to export a form, you should also add the data object corresponding to the form. This ensures that associated entity adapters can use the form.

Group Permissions

When exporting groups, group permissions on different data objects are also exported. However, when importing data, any permissions for missing data objects are ignored. If the group is exported as a way of exporting group permission setup, check the warnings carefully to make sure that permission requirements are met. For example, if a group has permissions for objects A, B, and C, but the target system only has objects A and B, the permissions for object C are ignored. If object C is added later, the group permissions for C must be added manually, or the group has to be imported again.

Report Permissions

When exporting groups that have permissions to view certain reports, ensure that the reports exist in the target environment. If reports are missing, consider removing the permissions before exporting the group.

Back Up the Database

Back up the database before importing data into a production system. This enables you to restore the data if anything goes wrong with the import. Backing up the database is always a good precaution before making significant changes.

Note: When you import forms and user-defined fields, you add entries to the database. These database entries cannot be rolled back or deleted. Before importing, be sure that the correct form version is active.

Import Data When the System Is Quiet

You cannot complete an import operation in a single transaction because the import includes schema changes. These changes affect currently running transactions on the system. To limit the impact of an import operation, separate the web application from general use and perform the operation when the system has the least activity, for example, overnight.

Updating the SDK Table

The SDK table contains metadata definitions for user-defined data objects. After you import data from an XML file into the SDK table, the values in the SDK_SCHEMA column may be modified with the schema name of the source system where the XML file was created. For this reason, after you import data from an XML file into the SDK table, you must check the schema name in the SDK_SCHEMA column, and if necessary, manually change it to the schema name on the target system where the Oracle Identity Manager database is running. To update the schema name in the SDK_SCHEMA column, you can run an SQL query similar to the following with SQL*Plus on Oracle Database installations or with SQL Query Analyzer on SQL Server installations:

```
UPDATE SDK SET SDK_SCHEMA='target system schema name'
```

If you do not update the schema name in the SDK_SCHEMA column, an error similar to the following may be generated when importing other XML files that modify user-defined field (UDF) definitions:

```
CREATE SEQUENCE UGP_SEQ
java.sql.SQLException: ORA-00955: name is already used by an existing object
```

2

Configuring Your Application Server

This chapter describes how to configure your application server. It contains these sections:

Configuring JBoss Application Server

This section describes how to configure JBoss application server. It contains these topics:

Removing JBoss Components

For a JBoss installation, some JBoss components are not required by Oracle Identity Manager. These files vary for stand-alone and cluster installations.

You can remove the following components:

- Cache invalidation service (keep for a clustered installation)
- J2EE client deployer service
- Integrated HAR deployer and Hibernate session Management services
- JMX Console
- Management console
- Console/email monitor alerts
- UUID key Generation
- JBoss scheduler manager
- Scheduler service
- Test queues and topics
- Mail service
- HTTP Invoker
- CORBA/IIOP
- AOP Application

- Web services support

Removing Files and Directories

The following sections list the files and directories that you can remove after installing Oracle Identity Manager.

Non-Clustered Installations

Remove the following files from *JBOSS_HOME*/server/default/deploy/:

- cache-invalidation-service.xml
- client-deployer-service.xml
- monitoring-service.xml
- scheduler-service.xml
- schedule-manager-service.xml
- http-invoker.sar
- mail-service.xml
- jboss-aop.deployer
- jboss-ws4ee.sar

Remove the following directories from *JBOSS_HOME*/server/default/deploy/:

- jboss-hibernate.deployer
- jmx-console.war
- management
- uuid-key-generator.sar

Remove the following file:

JBOSS_HOME/server/default/deploy/jms/jbossmq-destinations-service.xml.

Open the *JBOSS_HOME*/server/default/conf/jboss-service.xml file and remove the following attribute:

```
<attribute name'"RMI_IIOPService">jboss:service=CorbaORB</attribute>
```

Clustered Installations

Remove the following files from *JBOSS_HOME*/server/all/deploy/:

- client-deployer-service.xml
- monitoring-service.xml
- scheduler-service.xml
- schedule-manager-service.xml
- httpha-invoker.sar
- mail-service.xml
- jboss-aop.deployer
- jboss-ws4ee.sar

Remove the following directories from the *JBOSS_HOME*/server/all/deploy/:

- jboss-hibernate.deployer
- jmx-console.war
- management
- uuid-key-generator.sar

Remove the following file:

JBOSS_HOME/server/all/deploy-hasingleton/jms/jbossmq-destinations-service.xml

Open the *JBOSS_HOME*/server/all/conf/jboss-service.xml file and remove the following attribute:

```
<attribute name'"RMI_IIOPService">jboss:service=CorbaORB</attribute>
```

Configuring WebLogic Application Server

Set the transaction timeout parameter in the weblogic.profile file to 20 minutes or higher, depending on the size of your transactions.

3

Tuning Oracle Database for Oracle Identity Manager

As with any enterprise class business application, there is no simple procedure for tuning that works for all systems. This section describes one sample configuration and outlines principles for tuning.

Oracle Identity Manager has many configuration options. The best way to identify bottlenecks and optimize performance is to monitor key database performance indicators in your production environment and adjust the configuration from time to time. This chapter serves as a guideline to help you choose the initial baseline database configuration.

This chapter discusses the following topics:

Sample Instance Configuration Parameters

The following parameter settings are based on a server with 4 CPUs and 8GB RAM

Table 3–1 Sample Configuration Parameters

Parameter	Recommended Initial Settings for Oracle9*i*	Recommended Initial Settings for Oracle10*g*
`compatible`	9.2.0.0.0	10.2.0.2.0
`cursor_sharing`	EXACT	EXACT
`db_block_size`	8192	8192
`db_cache_size`	3200M	3200M
`db_keep_cache_size`	800M	800M
`log_buffer`	14289920	14289920
`shared_pool_size`	500M	500M
`db_file_multiblock_read_count`	16	16
`db_writer_processes`	2	2
`hash_join_enabled`	TRUE	Not applicable

Table 3–1 (Cont.) Sample Configuration Parameters

Parameter	Recommended Initial Settings for Oracle9*i*	Recommended Initial Settings for Oracle10*g*
java_pool_size	300M	300M
open_cursors	600	600
optimizer_feature_enable	9.2.0	10.2.0.2
optimizer_index_cost_adj	Between 0 and 20	Between 0 and 20
pga_aggregate_target	1100M	1100M
workarea_size_policy	AUTO	AUTO
processes	Set the processes parameter to the maximum number of concurrent users + the number of background processes + the number of SQL*PLUS and other user processes + 10 (as an extra cushion)	Set the processes parameter to the maximum number of concurrent users + the number of background processes + the number of SQL*PLUS and other user processes + 10 (as an extra cushion)
query_rewrite_enabled	TRUE	TRUE
query_rewrite_integrity	TRUSTED	TRUSTED
session_cached_cursors	300	300

Physical Data Placement

The out-of-the-box installation of Oracle Identity Manager uses only one physical tablespace to store database objects. Oracle Identity Manager database objects belong to one of the following categories:

- Physical tables
- Index
- Large objects (LOBS/CLOBS)

For better performance, create multiple locally managed tablespaces and store each category of database object in a dedicated tablespace. This optimizes storage for efficient data access. Oracle recommends that you place the following User Profile Audit (UPA) component tables and indexes in their own tablespaces:

- UPA
- UPA_FIELDS
- UPA_GRP_MEMBERSHIP
- UPA_RESOURCE
- UPA_USR

These tables can store significant amounts of historical data and can be used by historical reports.

The database schema includes the following tables for reconciliation data:

- RCA
- RCB
- RCD

- RCE
- RCH
- RCM
- RCP
- RCU
- RPC

If your environment generates a large amount of reconciliation data, move these tables to a new tablespace. Use the locally managed tablespaces with automatic extent allocation.

You can use performance metrics to identify tables that are accessed frequently, or *hot* tables. To reduce I/O contention, move hot tables to dedicated tablespaces. See "Database Performance Monitoring" on page 3-4 for more on performance metrics.

Pinning Sequences and Stored Procedures in System Global Area (SGA)

Oracle Identity Manager uses sequence objects to generate unique record identifiers. Oracle Identity Manager also uses stored procedures to perform specific database operations. To optimize performance in production, pin the sequence objects and stored procedures in SGA. A script named create_db_trigger.sql is shipped with the Oracle Identity Manager installation for this purpose. The create_db_trigger.sql script is written for the Oracle Identity Manager database account `SYSADM`. It is a sample Oracle login account.

This script is located in the following installation directory:

\installServer\Xellerate\db\oracle

To pin the sequence objects and stored procedures:

1. Log in as `SYS`.
2. Start Oracle SQL*Plus (the Oracle client tool), at a command prompt, by typing the following command:

   ```
   sqlplus /nolog
   ```

3. Connect to the Oracle instance as `SYS` user with `SYSDBA` role.

   For example, type the following command:

   ```
   CONNECT SYS/sys_password@db_instance AS SYSDBA
   ```

   Where *`sys_password`* is the password for the `SYS` user account, and *`db_instance`* is the Net 8 service name for connecting to the database instance.

   For example:

   ```
   CONNECT SYS/sys@xeltest AS SYSDBA
   Connected.
   ```

4. Edit the create_db_trigger.sql script in a text editor, and specify your actual Oracle Identity Manager database account name.
5. In create_db_trigger.sql, substitute all references to sysadm with the account name you actually used.

   For example, if your Oracle Identity Manager database account name is `myschema`, edit your script as follows:

```
create or replace trigger cache_seq after startup on database begin
myschema.pin_obj;
-- pin all sysadm's sequences in shared_pool
myschema.pin_sp;
-- pin all commonly executed XELL stored procedures/functions
end;
/
```

6. Run the create_db_trigger.sql script.

   This script creates a database trigger that is fired every time the database starts up. Any subsequent database bounces automatically pin the sequences and stored procedures in SGA.

7. While connected to Oracle as the `SYS` user, run the following commands:

   ```
   EXEC database_user.PIN_OBJ;
   EXEC database_user.PIN_SP;
   ```

   Where *`database_user`* is the database account.

   Run these commands only once during initial schema creation. Bouncing the Oracle server is not required.

Database Performance Monitoring

To identify performance bottlenecks, you can monitor real-time performance metrics for the Oracle Identity Manager database.

Perform the following at regular intervals:

- Monitor real-time performance using a performance-monitoring tool such as Statspack in Oracle9i or Automatic Workload Repository (AWR) in Oracle10*g*.
- Collect initial schema statistics upon implementation of Oracle Identity Manager.

  Update schema statistics regularly, so that the Cost-Based Optimizer (CBO) has access to the most up-to-date statistics.

  This helps the CBO determine a efficient query execution plan that is based on the current state of data.
- Look for relevant recommendations provided in advisory sections in the Oracle Statspack report or AWR report, and adjust the instance configuration parameters according to the advised settings.

4

Managing the Cache

Oracle Identity Manager uses two types of caching: **global** and **ThreadLocal**.

The global cache stores information globally. Any part of the system can access information that is stored in this cache. The global cache uses OSCache from OpenSymphony. One advantage of using OSCache is its support for cluster environments. Database queries are usually stored in the global cache so that repeated queries are not run against the database again.

The ThreadLocal cache stores information that is used multiple times in a single transaction. For example, a query that is issued many times during a transaction uses data from the ThreadLocal cache. The data used for this query does not change for the transaction.

Oracle Identity Manager allows caching by category. You can enable and disable caching for specific entities and configure separate expiration times.

This chapter discusses the following topics:

Sample Cache Configuration

Example 4–1 is a snippet from the `Cache` section in the xlconfig.xml file:

Example 4–1 xlconfig.xml Snippet

```
<Cache>
  <Enable>false</Enable>
  <ThreadLocalCacheEnabled>false</ThreadLocalCacheEnabled>
  <ExpireTime>14400</ExpireTime>

  <CacheProvider>com.thortech.xl.cache.OSCacheProvider</CacheProvider>
  <XLCacheProvider>
    <Size>5000</Size>
    <MultiCastAddress>231.121.212.133</MultiCastAddress>
  </XLCacheProvider>

  <!-- Individual cache categories -->
```

```
  <!-- Adapters and event handlers to be executed on update/insert/delete -->
  <DataObjectEventHandlers>
    <Enable>false</Enable>
    <ExpireTime>14400</ExpireTime>
  </DataObjectEventHandlers>

...
...
...
</Cache>
```

Note: Oracle recommends that you disable caching in development environments. Data in development environments changes frequently. If cached data is not refreshed in time, it can cause problems for developers working with the product.

General Cache Configuration Properties

The `Cache` tag refers to the cache configuration and what is contained between the beginning and the end `Cache` tag. Table 4–1 describes the entries in the `Cache` section:

Table 4–1 Cache Configuration Parameters

Property	Description
`Enable`	Enables components in the cache configuration for categories that are not explicitly defined in the configuration file. If the configuration file does not contain a particular category, the cache uses this entry to enable or disable the category.
`ThreadLocalCacheEnabled`	Enables or disables ThreadLocal caching.
`ExpireTime`	Specifies a default expiration time for components in the cache configuration.
`CacheProvider`	Identifies the complete class path of the provider used for caching. Do not change this property.
`XLCacheProvider`	Specifies cache provider properties. In Example 4–1, the `Size` and `Multicast Address` properties are specified.
`XLCacheProvider - Size`	Specifies the size of the cache. This number reflects the number of items that the cache stores. If the size is reached, new items are stored in the cache while the least used are pushed out of the cache.
`XLCacheProvider - MultiCastAddress`	Used for multicast communication among all of Oracle Identity Manager components.

Note: The same MultiCast Address must be used for all Oracle Identity Manager installations in an environment, for example, for all the nodes in a cluster. Cache flushes are propagated to all installations using MultiCast IP. If multicasting is disabled, cache flush is not possible.

Category-Based Cache Configuration Properties

After you preform general cache configuration, each component or category is shown with its own tag name. The tag name reflects a category name that is used in the code to store information in the cache. You can enable or disable each category independently of other categories, and you can set the expiration time for each component or category.

Table 4–2 lists the categories in the cache configuration file. By default, all the categories are disabled in the cache configuration file unless otherwise mentioned in Table 4–2.

Table 4–2 Category Based Cache Configuration Parameters

Category Name	Description
`DataObjectEventHandlers`	List of event handlers to be run when data object changes occur. This is the location where custom event handler and entity adapters are attached to a data object.
`ProcessDefinition`	Process definition information, for example process attributes, tasks, task mappings, and so on.
`RuleDefinition`	Rule definition information.
`FormDefinition`	Form definition information.
`ColumnMap`	DB column name from a column code. This is enabled by default.
`UserDefinedColumns`	User Defined Form and column definitions
`ObjectDefinition`	Object definition information.
`StoredProcAPI`	Used to stored total counts when calling APIs with paging capability. Because information changes frequently, the default expiration time for this category is 10 minutes.
`NoNeedToFlush`	This category defines data that does not need to be flushed and does not fall into a particular category. This category does not have an expiration time. This information is typically populated during initial database setup and never changes in an installation.
`MetaData`	DB field metadata information. This is category is enabled by default.
`AdapterInformation`	Adapter variables, compilation status and so on.
`OrgnizationName`	Cache organization names.
`Reconciliation`	Reconciliation rules.
`SystemProperties`	Caches system properties.
`LookupDefinition`	Caches the conversions between lookup names and fields.
`UserGroups`	Caches user groups.
`LookupValues`	Caches the lookup values for a given lookup name.
`ITResourceKey`	IT Resources DB key cache.
`ServerProperties`	Caches what data is to be encrypted along with System Properties
`ColumnMetaData`	Database metadata information for common queries.
`CustomResourceBundle`	Caches custom resource bundle.
`CustomDefaultBundle`	Caches custom default bundle.

Table 4–2 (Cont.) Category Based Cache Configuration Parameters

Category Name	Description
`ConnectorResourceBundle`	Caches connector resource bundles
`EmailDefinition`	Caches e-mail definition information
`LinguisticSort`	Caches database sort parameters
`RecordExists`	Caches user keys
`GenericConnector`	Caches pertinent data about a particular Generic Technology Connector instance
`GenericConnectorProviders`	Caches the provider parameter values associated with a particular Generic Technology Connector instance

Class Reloading

Class reloading refers to automatically reloading classes without restarting the server. Class reloading settings are useful for scheduled tasks and adapter-related files. Oracle recommends that you enable reloading in development environments. You must restart the Oracle Identity Manager server if cache reloading is disabled and any new adapters are imported, existing adapters are changed, or any .jar files are modified.

> **Note:** Oracle recommends that you disable class reloading in production environments to improve performance.

The class reloading configuration information is included in the xlconfig.xml file as follows:

```
<ClassLoading>
  <ReloadEnabled>true</ReloadEnabled>
  <ReloadInterval>15</ReloadInterval>
  <LoadingStyle>ParentFirst</LoadingStyle>
</ClassLoading>
```

- `ReloadEnabled` enables class reloading on regular basis.
- `ReloadInterval` specifies the time to reload (in seconds).
- `LoadingStyle` specifies the type of loading used.

  The following are the different types of loading:

  - `ParentFirst` looks for the classes in the parent before loading them from the jar files in ADPClassLoader classpath.
  - `ParentLast` overrides the classes from the parent. Using `ParentLast` may cause `ClassCastExceptions`.
  - `ParentLoader` is the ThreadContext Class Loader.

Purging the Cache

If you want to purge the cache before the allocated amount of time, use the `PurgeCache` utility in the *XL_HOME*/bin directory. This utility purges all elements in the cache.

Depending on the platform, the PurgeCache utility is a batch file or a shell script. After you edit the XEL_HOME and JAVA_HOME environment variables to point to the correct location, you can run the PurgeCache from the command line.

To use the PurgeCache utility, run PurgeCache.bat *category name* on Windows systems or PurgeCache.sh *category name* on UNIX/Linux systems. The category name argument represents the name of the category that needs to be purged. For example, the following commands purge all FormDefinition entries from a system and its clusters:

```
PurgeCache.bat FormDefinition
PurgeCache.sh FormDefinition
```

To purge all Oracle Identity Manager categories, pass a value of "ALL" to the PurgeCache utility.

Note:

- The category name argument of the PurgeCache utility is case sensitive.
- A java.lang.NullPointerException is thrown after running this script. However, this exception does not prevent data from being purged.

Optimal Cache Configuration for a Production Environment

Post-deployment changes to the cache configuration may affect performance and usage. Configure your cache using utmost caution.

The following are guidelines for configuring the Oracle Identity Manager cache for a production environment:

- Set all properties to true, except for the <StoredProcAPI> setting.
- Increase the <XLCacheProvider> size to 15000 (default value is 5000).

  Example 4–2 shows the recommended values for the Oracle Identity Manager cache configuration file (xlconfig.xml) in a production environment.

Example 4–2 Recommended Cache Values for xlconfig.xml in a Production Environment

```
<Cache>
  <Enable>true</Enable>
  <ThreadLocalCacheEnabled>true</ThreadLocalCacheEnabled>
  <ExpireTime>14400</ExpireTime>
  <CacheProvider>com.thortech.xl.cache.OSCacheProvider</CacheProvider>
  <XLCacheProvider>
    <Size>15000</Size>
    <MultiCastAddress>231.172.169.176</MultiCastAddress>
  </XLCacheProvider>

<!-- Individual cache categories -->
<!-- Adapters and event handlers to be executed on update/insert/delete -->
  <DataObjectEventHandlers>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </DataObjectEventHandlers>
  <ProcessDefinition>
    <Enable>true</Enable>
```

```
    <ExpireTime>14400</ExpireTime>
  </ProcessDefinition>
  <RuleDefinition>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </RuleDefinition>
  <FormDefinition>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </FormDefinition>
  <ColumnMap>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </ColumnMap>
  <UserDefinedColumns>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </UserDefinedColumns>
  <ObjectDefinition>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </ObjectDefinition>
  <StoredProcAPI>
    <Enable>false</Enable>
    <ExpireTime>600</ExpireTime>
  </StoredProcAPI>

<!-- This information never needs to flushed out. For example, key for requests
organization and so on. -->

  <NoNeedToFlush>
    <Enable>true</Enable>
    <ExpireTime>-1</ExpireTime>
  </NoNeedToFlush>

<!-- Metadata Information -->
  <MetaData>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </MetaData>

<!-- Adapter Mapping Information -->
  <AdapterInformation>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </AdapterInformation>

<!-- Name of the organization for a given key and vice versa -->
  <OrgnizationName>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </OrgnizationName>

<!-- Reconciliation rules -->
  <Reconciliation>
    <Enable>true</Enable>
    <ExpireTime>14400</ExpireTime>
  </Reconciliation>

<!-- System Properties -->
```

```
    <SystemProperties>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </SystemProperties>
    <LookupDefinition>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </LookupDefinition>
    <UserGroups>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </UserGroups>
    <LookupValues>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </LookupValues>
    <ITResourceKey>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </ITResourceKey>
    <RecordExists>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </RecordExists>
    <ServerProperties>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </ServerProperties>

  <!-- Column Meta Data -->
    <ColumnMetaData>
      <Enable>true</Enable>
      <ExpireTime>14400</ExpireTime>
    </ColumnMetaData>
    <CustomResourceBundle>
      <Enable>true</Enable>
      <ExpireTime>-1</ExpireTime>
    </CustomResourceBundle>
    <CustomDefaultBundle>
      <Enable>true</Enable>
      <ExpireTime>-1</ExpireTime>
    </CustomDefaultBundle>
    <ConnectorResourceBundle>
      <Enable>true</Enable>
      <ExpireTime>-1</ExpireTime>
    </ConnectorResourceBundle>
    <LinguisticSort>
        <Enable>true</Enable>
        <ExpireTime>-1</ExpireTime>
    </LinguisticSort>
    <GenericConnector>
        <Enable>true</Enable>
        <ExpireTime>-1</ExpireTime>
    </GenericConnector>
    <GenericConnectorProviders>
        <Enable>true</Enable>
        <ExpireTime>-1</ExpireTime>
    </GenericConnectorProviders>
  </Cache>
```

5

Securing Your Deployment

This chapter describes how to use Oracle Application Server Single Sign-On to secure your Oracle Identity Manager deployment.

This chapter discusses the following topics:

- Securing the Administrative and User Console
- Securing the Self Registration and Change Password Pages

> **See Also:** See the *Oracle Application Server Single Sign-On Administrator's Guide* for information on how to protect URLs.

Securing the Administrative and User Console

To secure the Administrative and User Console, use Oracle Application Server Single Sign-On to protect the following URLs:

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

Securing the Self Registration and Change Password Pages

To secure the Self Registration and Change Password pages, use Oracle Application Server Single Sign-On to protect the following URL:

`http://hostname:port/xlWebApp/Logon.do`

After using Oracle Application Server Single Sign-On to protect the Self Registration and Change Password pages, you can use the following URLs to directly access the pages:

`http://hostname:port/xlWebApp/selfRegister.do?method=New%20Registration`
`http://hostname:port/xlWebApp/forgetPassword.do?method=displayVerifyUserId`

6

Integrating with Oracle Access Manager

This chapter describes using Oracle Access Manager to manage user authentication and authorization when a user logs in to Oracle Identity Manager.

This chapter covers the following topics:

> **Note:** While this chapter focuses on using JBoss as the application server in the integration, the same configuration steps apply to instances where Oracle Identity Manager is deployed on WebSphere, WebLogic or any other J2EE application server that is supported by Oracle Identity Manager.

About the Integration with Oracle Identity Manager

The integration of Oracle Access Manager with Oracle Identity Manager provides a secure Web-based infrastructure for identity management for all customer applications and processes. Oracle Access Manager integrates identity and access management across Oracle Identity Manager, enterprise resources, and other domains deployed on eBusiness networks. Oracle Access Manager provides the foundation for managing the identities of customers, partners, and employees across Internet applications. These user identities are combined with security policies for protected Web interaction.

This integration adds the following features to Oracle Identity Manager implementations:

- **Oracle Access Manager authentication, authorization, and auditing** services for Oracle Identity Manager.
- **Oracle Access Manager single sign-on** for Oracle Identity Manager and other Oracle Access Manager-protected resources within a single domain or across multiple domains.

- **Oracle Access Manager authentication schemes,** the following schemes provide single sign-on for Oracle Identity Manager:
  - **Basic**: Users must enter a user name and password in a window supplied by the Web server.

    This method can be redirected to SSL.'
  - **Form:** This method is similar to the basic challenge method, but users enter information in the custom HTML form.

    You can choose the information users must provide in the form that you create.
  - **X509 Certificates**: X.509 digital certificates over SSL.

    A user's browser must supply a certificate.
  - **Integrated Windows Authentication (IWA)**: Users will not notice a difference between an Oracle Access Manager authentication and IWA when they log on to the desktop, open an Internet Explorer (IE) browser, request a Oracle Access Manager-protected Web resource, and complete single sign-on.
  - **Custom**: Additional forms of authentication can be incorporated through use of the Oracle Access Manager Authentication Plug-in API.
- **Session timeout**: Oracle Access Manager enables you to set the length of time that a user session is valid.
- **Ability to use the Oracle Access Manager Identity System**: This system provides identity management features such as user self-service for registration and updating user profiles, portal inserts, delegated administration, and workflows. You can send Identity System data to back-end applications using a custom data template and a workflow.

Integration Architecture

Oracle Identity Manager has two authentication mechanisms:

- Default mode, where Oracle Identity Manager manages the credential validation and session maintenance.
- Single sign-on mode, where Oracle Identity Manager looks for an HTTP header variable that is passed to it.

  The header variable should contain the user ID of the Oracle Identity Manager user.

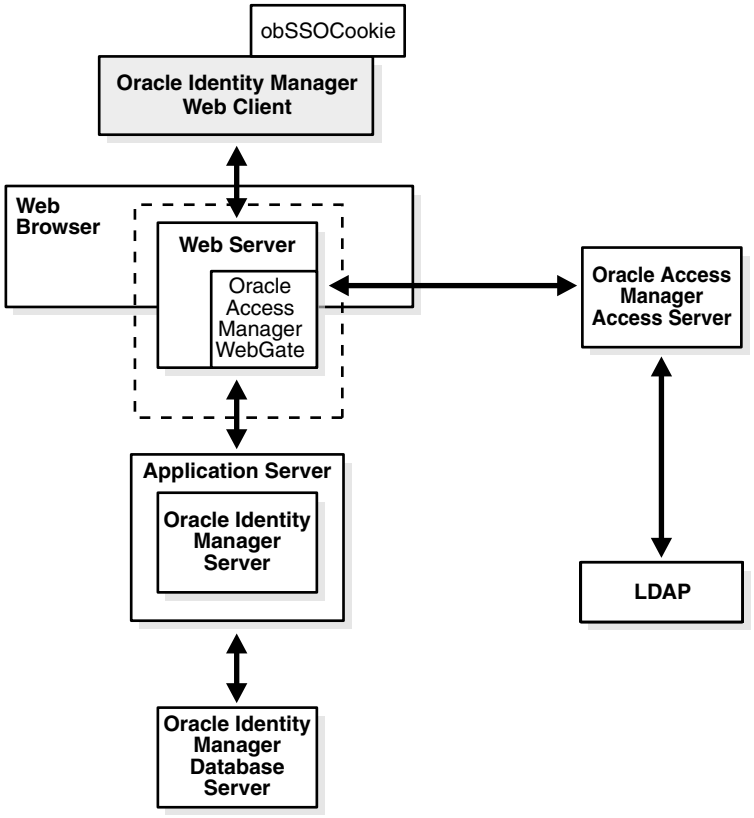
Oracle Access Manager single sign-on with Oracle Identity Manager is achieved as follows:

- Deploy an HTTP Server in front of the J2EE Application server.
- Deploy the HTTP Server as a reverse proxy.
- Deploy a WebGate on the HTTP Server.
- Populate a header variable with an attribute value that is stored in the LDAP directory used by Oracle Access Manager.
- Configure IOracle Identity Manager to use the single sign-on mode of authentication.

Figure 6–1 shows the architecture for single sign-on between Oracle Identity Manager and Oracle Access Manager.

The user accesses the Administrative and User Console with a Web browser. The WebGate intercepts the user's HTTP request and checks for the presence of an obSSOCookie. If the cookie does not exist or it has expired, the user is challenged for credentials. Oracle Access Manager verfies the credentials, and if the user is authenticated, the WebGate redirects the user to the requested resource and passes the required header variable to Oracle Identity Manager. Oracle Identity Manager, which has been configured to read a HTTP Header variable instead of its authentication, reads the HTTP Header and uses the value stored in the variable as the logged in user.

Figure 6–1 Integration with Oracle Identity Manager



Process overview: Single sign-on with Oracle Identity Manager

1. A user attemps to access the Administrative and User Console.
2. A WebGate that is deployed on the HTTP server intercepts the request.
3. The WebGate checks the Access Server to determine if the resource (the Oracle Identity Manager URL) is protected.

   The security policy in the Access System contains an authentication scheme, authorization rules, and allowed operations based on authentication and authorization success or failure.
4. If a valid session does not exist, and the resource is protected, WebGate prompts the user for credentials.
5. If the credentials are validated, Oracle Access Manager performs the actions that are defined in the security policy for the resource and sets an HTTP header variable that maps to the Oracle Identity Manager user ID.

6. If a vali session cookie exists, and if the user is authorized to access the resource, WebGate redirects the user to the requested Oracle Identity Manager resource.
7. The Administrative and User Console reads the HTTP header variable and sets the value as the logged-in user.
8. The Administrative and User Console generates the applications pages, pending any further authorization checks performed in Oracle Identity Manager.

Preparing Your Environment

Complete the following to prepare your environment for the integration.

Task overview: Preparing your environment for the integration

1. Install a supported directory server according to vendor instructions.
2. Install and configure Oracle Access Manager using the directory server as the LDAP repository.
3. Ensure that the Oracle Identity Manager J2EE application server is proxied by an HTTP server.
4. Configure the Web browser to allow cookies, according to vendor instructions.
5. Set up Oracle Access Manager for Oracle Identity Manager.

   See "Setting Up Oracle Access Manager Single Sign-On for Oracle Identity Manager" on page 6-4 for details.

Setting Up Oracle Access Manager Single Sign-On for Oracle Identity Manager

The following procedures describes setting up WebGate on an HTTP server and configuring Oracle Access Manager for single sign-on with Oracle Identity Manager.

Note that you can configure form-based authentication for logins that use either ASCII or non-ASCII characters. Due to browser limitations, Basic authentication schemes only accept ASCII login credentials.

> **See also:** For more information about configuring authentication and authorization in Oracle Access Manager, see the *Oracle Access Manager Access Administration Guide*.

To set up a WebGate on an HTTP server

1. Install and configure Oracle Access Manager on a supported platform, using a supported LDAP server.

   See the *Oracle Access Manager Installation Guide* for details.
2. Install a WebGate on the Oracle Identity Manager HTTP server.

   Do not install the WebGate against an application server that supports HTTP services, for example, BEA Weblogic. If your application server is JBoss, IBM WebSphere, or BEA Weblogic, install an HTTP server such as Apache, iPlanet, or Oracle HTTP Server.
3. Configure the HTTP server to forward user requests to the J2EE application server and forward responses from the Oracle Identity Manager back to the user.

To configure single sign-on in Oracle Access Manager

1. In the landing page for the Access System, click the link for the Policy Manager, and click Create Policy Domain.
2. Create a policy domain and policies to restrict access to the Oracle Identity Manager URLs.
3. In the Access System Console, define host identifiers for Oracle Identity Manager.
4. Click the link for the Policy Manager, click the link for the Oracle Identity Manager policy domain, click the Resources tab, and define resources for Oracle Access Manager to protect.
5. Click the Authorization Rules tab and define an authorization rule to determine which authenticated users can access the Oracle Identity Manager URLs.
6. Click the Default Rules tab.

   The Authentication Rule sub-tab is selected.
7. Define an authentication rule, for example, Basic Over LDAP.
8. Click the Actions sub-tab and define an authorization action that sets a custom HTTP header variable upon successful authorization.

   The header variable should contain a value that maps to the Oracle Identity Manager user ID.
9. Click the Policies tab, click Add, and define an access policy in the Oracle Identity Manager policy domain and add the Oracle Identity Manager URL resources to this policy.

Setting Up Oracle Identity Manager for Single Sign-On with Oracle Access Manager

The following procedure describes how to set up Oracle Identity Manager for integration with Oracle Access Manager.

To configure single sign-on for Oracle Identity Manager

1. Stop the application server gracefully.
2. Launch a plain-text editor and open the following file:

   <*XL_HOME*>\xellerate\config\xlconfig.xml
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 as follows.

   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.

   Refer to your application and web server vendor documentation for details.

6. Restart the application server.

Configuring Apache as a Proxy for JBoss

TheAdministrative and User Console runs in a J2EE application server, for example, JBoss, BEA Weblogic, and IBM WebSphere. You cannot install an AccessGate directly against these application servers. You can deploy a Web servre, for example, Apache, Oracle HTTP Server, and iPlanet in front of these application servers. You can deploy the AccessGate on the Web server, and configure the Web server to route requests to the Oracle Identity Manager application and forward responsees back to the user.

For application servers such as JBoss, you must deploy an additional plug-in, referred to as the mod_jk plug-in or the JBoss plug-in, on the Web server. You can obtain the mod_jk plug-in from the Apache Tomcat Web site, under the Tomcat connectors section. As of the time of publication, the URL as follows:

http://tomcat.apache.org/download-connectors.cgi

To configure the Apache HTTP server as a proxy for JBoss

1. Download and install a version of the Apache HTTP Server that is supported by Oracle Access Manager.
2. Download the latest stable version of the Jakarta (also known as Tomcat) mod_jk plug-in from the following URL:

   http://tomcat.apache.org/download-connectors.cgi
3. Extract the file and rename it to mod_jk.so.
4. Copy this file to the following directory:

   Apache_install_dir\modules
5. Create the following text files in the directory *Apache_install_dir*\conf:
   - mod-jk.conf
   - workers.properties
   - uriworkermap.properties

   Oracle recommends that you do not rename uriworkermap.properties and workers.properties. If you do, your configuration may stop working. The locations of these files are defined under two registry keys: worker_file and worker_mount_

file. These files are in HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector*version_number*.

6. Copy the following configuration into the mod-jk.conf file:

```
# Load mod_jk module
# Specify the file name of the mod_jk lib
LoadModule jk_module modules/mod_jk.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties

# Where to put jk logs
JkLogFile logs/mod_jk.log

# Set the jk log level [debug/error/info]
JkLogLevel info

# Select the log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y]"

# JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories

# JkRequestLogFormat
JkRequestLogFormat "%w %V %T"

# Mount  your  applications
JkMount /application/* loadbalancer

# You can use external file for mount points.
# It will be checkded for updates each 60 seconds.
# The format of the file is: /url=worker
# /examples/*=loadbalancer
JkMountFile conf/uriworkermap.properties

# Add shared memory.
# This directive is present with 1.2.10 and
# later versions of mod_jk, and is needed for
# for load balancing to work properly
JkShmFile logs/jk.shm

# Add jkstatus for managing runtime data
<Location /jkstatus/>
JkMount status
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Location>
```

7. Copy the following into the workers.properties file:

```
# Define the list of workers that will be used
# for mapping requests
worker.list=loadbalancer

# Define node1
worker.node1.port=8009
worker.node1.host=<Put your Identity Manager App Server FQDN name here>
worker.node1.type=ajp13
worker.node1.lbfactor=1
```

```
worker.node1.local_worker=1 (1)
worker.node1.cachesize=10
#Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1
worker.loadbalancer.sticky_session=1
worker.loadbalancer.local_worker_only=1
```

8. Copy the following into the uriworkermap.properties file.

   Configure the mapping according to the `worker.list` entry defined in the workers.properties file. This is not always `loadbalancer`, although this is shown in the following example:

```
# Simple worker configuration file
# Mount the servlet context to the ajp13 worker
/jmx-console=loadbalancer
/jmx-console/*=loadbalancer
/web-console=loadbalancer
/web-console/*=loadbalancer
/xlWebApp=loadbalancer
/xlWebApp/*=loadbalancer
/Nexaweb=loadbalancer
/Nexaweb/*=loadbalancer
```

7

Integrating with Oracle Application Server Single Sign-On

This chapter describes how to use Oracle Application Server (OracleAS) Single Sign-On, a component of OracleAS, to manage user authentication and authorization when a user logs in to Oracle Identity Manager.

> **See Also:** The *Oracle Application Server Single Sign-On Administrator's Guide* for more information about deploying OracleAS Single Sign-On.

This chapter assumes you are familiar with OracleAS Single Sign-On and Oracle Identity Management infrastructure, and that you have the required components, including your application server, web server, Oracle Identity Manager, OracleAS Single Sign-On, and Oracle Internet Directory, already installed.

> **Important:** Several different configurations, including application and web servers, are possible in an Oracle Identity Manager and OracleAS Single Sign-On environment.
>
> To demonstrate one possible configuration, this chapter describes how to integrate with OracleAS Single Sign-On using the Oracle Containers for J2EE (OC4J) application server and the Oracle Application Server OC4J Internet Information Server (IIS) Plugin plugin. The information in this chapter is based on IIS version 6.0.
>
> Refer to your application and web server vendor's documentation for more information about configuring single sign-on.

This chapter covers the following topics:

- Setting Up OC4J IIS Plugin to Communicate with OracleAS Single Sign-On
- Setting Up Oracle Identity Manager for Single Sign-On with OracleAS Single Sign-On
- Creating Single Sign-On User Accounts for Oracle Identity Manager Users

Setting Up OC4J IIS Plugin to Communicate with OracleAS Single Sign-On

You must install and configure the Oracle Application Server OC4J Plugin, which is an IIS plugin for OC4J, so that the OC4J application server can communicate with the

OracleAS Single Sign-On server. The Oracle Application Server OC4J Plugin is a file named opii.dll.

Perform the following steps to install and configure the Oracle Application Server OC4J Plugin:

1. Download the Oracle Application Server OC4J Plugin from the Oracle Technology Network (OTN) using the following steps:
   a. Go to the OTN Web site at the following URL:

      `http://www.oracle.com/technology/index.html`

   b. Click **Downloads** on the horizontal navigation menu at the top of the page.
   c. Scroll to the **Middleware** section of the page and click **SOA Suite** in the **Developer Tools** section.
   d. Click **See All** in the **Oracle SOA Suite 10g Release 3 (10.1.3.1.0)** section.
   e. Expand the **Oracle SOA Suite 10g Companion CD** entry and you will see the Oracle Application Server OC4J Plugin listed as a component.
   f. Download CD1 for the Oracle SOA Suite 10g Companion CD by clicking **CD1** for the appropriate operating system.
2. Open your Registry Editor and perform the following steps:

   Note: This procedure uses example steps using regedit.

   a. Click **HKEY_LOCAL_MACHINE**, then click **SOFTWARE**, then right-click **Oracle** and select **New**, then select **Key**, and name it opii.
   b. Right-click the opii entry, select **New**, then select **String Value** and name the String Value log_file.
   c. Right-click the log_file entry and select Modify. The Edit String dialog box appears.
   d. Enter a path in the **Value data** field to location where you want to keep the opii log file and click **OK**.
   e. Right-click the opii entry, select **New**, then select **String Value** and name the String Value log_level. This log_level string value specifies the desired log level for opii, for which debug, inform, error, and emerg are valid values.
   f. Right-click the opii entry, select **New**, then select **String Value** and name the String Value server_defs.
   g. Right-click the server_def String Value and select **Modify**. The Edit String dialog box appears.
   h. Enter a path to the location where the opii.conf file will reside. You will create the opii.conf file in step 9.
3. Start the IIS Management Console, then expand the entry for the node hosting the IIS server that will communicate with the OracleAS Single Sign-On server, then expand the **Web Sites** entry, then right-click the **Default Web Sites** entry and select **New**, then select **Virtual Directory**. The Virtual Directory Creation Wizard appears. Click **Next** and perform the following steps:
   a. Enter opii in the **Alias Name** field and click **Next**.

b. Enter the location where the opii.dll file is located in the **Path** field and click **Next**.

c. Select the **Read**, **Run scripts**, and **Execute** options on the Virtual Directory Access Permissions screen and click **Next**. Click **Finish** to close the Virtual Directory Creation Wizard.

4. Add the opii.dll Oracle Application Server OC4J Plugin as a filter to your IIS web sites using the following steps:

a. In the IIS Management Console, right-click the **Default Web Sites** entry and select **Properties**. The Default Web Site Properties dialog box appears.

b. Click the **ISAPI Filters** tab and click **Add**.

c. Enter opii in the **Filter Name** field.

d. Enter the path to the location of the opii.dll Oracle Application Server OC4J Plugin in the **Executable** field.

e. Click **OK** on the Add/Edit Filter Properties dialog box. Click **OK** on the Default Web Site Properties dialog box.

5. Give permission to the IIS group on the <*OSSO_HOME*>\bin folder using the following steps:

a. Right-click the <*OSSO_HOME*>\bin folder and select **Properties**.

b. Click the **Security** tab.

c. Add the IIS_WPG group with Read and Execute permissions.

6. Restart the IIS server using the following steps from the IIS Management Console:

a. Right-click the node hosting the IIS server that will communicate with the OracleAS Single Sign-On server, select **All Tasks**, and then select **Restart IIS**. The Stop/Start/Restart dialog box appears.

b. Select **Restart <*the name of IIS server*>** and click **OK**.

c. After the IIS server restarts, verify the opii.dll Oracle Application Server OC4J Plugin is running by right-clicking **Default Web Sites**, selecting **Properties**, selecting the **ISAPI Filters** tab, and confirming there is a green arrow pointing up for the opii filter.

7. On the IIS Management Console, click **Web Services Extensions**, select **opii**, and then click the **Allow** button.

8. Identify the port for the ajp13 protocol using the following steps:

a. On the machine hosting the OC4J application server, open the <*OC4J_HOME*>\j2ee\<*OC4J_INSTANCE*>\config\default-web-site-.xml file in a text editor.

Note: <*OC4J_HOME*> represents the location where OC4J is installed. <*OC4J_INSTANCE*> represents the name of the OC4J instance.

---

b. Search for the string ajp13.

c. Identify the port number for ajp13, for example 8889.

9. Create a file named opii.conf in the opii directory that contains the following entries. The entries list the Oracle Identity Manager applications protected by

OracleAS Single Sign-On, the name of the machine hosting Oracle Identity Manager (for example, *host_name*), and the port number for ajp13 (for example, *ajp13 port number*):

```
Oc4jMount /xlWebApp ajp13://host_name:ajp13 port number
Oc4jMount /xlWebApp/* ajp13://host_name:ajp13 port number
Oc4jMount /xlScheduler ajp13://host_name:ajp13 port number
Oc4jMount /xlScheduler/* ajp13://host_name:ajp13 port number
Oc4jMount /Nexaweb ajp13://host_name:ajp13 port number
Oc4jMount /Nexaweb/* ajp13://host_name:ajp13 port number
```

Setting Up Oracle Identity Manager for Single Sign-On with OracleAS Single Sign-On

Perform the following steps to set up Oracle Identity Manager for integration with OracleAS Single Sign-On:

1. Stop the application server gracefully.
2. Launch a plain-text editor and open the following file:

   <*XL_HOME*>\xellerate\config\xlconfig.xml

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 as follows.

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

5. Restart the application server.

Creating Single Sign-On User Accounts for Oracle Identity Manager Users

You must create an entry in Oracle Internet Directory for each Oracle Identity Manager user that will use OracleAS Single Sign-On for authentication. Oracle Internet Directory is the repository for all OracleAS Single Sign-On user accounts and passwords. The OracleAS Single Sign-On server authenticates users against their entries in Oracle Internet Directory.

Perform the following steps to create an entry in Oracle Internet Directory for each Oracle Identity Manager user that will use OracleAS Single Sign-On for authentication:

1. Log in to the Oracle Delegated Administration Services home page at the following URL:

   http://*host*:*port*/oiddas/

   host represents the name of the computer where Oracle Delegated Administration Services is located, and *port* is the port number of this server. Oracle Delegated Administration Services and OracleAS Single Sign-On generally have the same host name.

2. Click the **Directory** tab.
3. Click **Create** on the **Users** tab.
4. Create the information about the Oracle Identity Manager user by entering information in the following fields:
   - **First Name**
   - **Last Name**
   - **User ID**

     Note: The User's ID must be the same as User's ID for Oracle Identity Manager.

   - **e-mail**
   - **Password** for OracleAS Single Sign-On (and confirm by entering it twice)
5. Create the user by clicking the Submit button.

8

Using the Reconciliation Archival Utility

This chapter describes how to use the Reconciliation Archival utility. It contains the following topics:

Understanding the Reconciliation Archival Utility

Oracle Identity Manager stores reconciliation data from target systems in the following tables, which are called called active reconciliation tables:

- RCA
- RCB
- RCD
- RCE
- RCH
- RCM
- RCP
- RCU
- RPC

During the reconciliation process, Reconcilation Manager reconciles data in the active reconcilation tables with Oracle Identity Manager's core tables. Because Reconcilation Manager does not remove reconciled data from the active reconcilation tables, they may eventually grow very large, resulting in decreased performance during the reconciliation process. You can use the Reconciliation Archival utility to archive data that has been reconciled with Oracle Identity Manager. The Reconciliation Archival utility stores archived data in the following tables, called archive reconciliation tables, which have the same structure as the active reconcilation tables:

- ARCH_RCA
- ARCH_RCB
- ARCH_RCD

- ARCH_RCE
- ARCH_RCH
- ARCH_RCM
- ARCH_RCP
- ARCH_RCU
- ARCH_RPC

You can use the Reconciliation Archival utility to perform the following tasks:

- Archive all data or specified from the active reconciliation tables to the archive reconciliation tables
- Delete data from the archive reconciliation tables
- Delete data from the active reconciliation tables

When you archive selective data from the active reconciliation tables to the archive reconciliation tables, you must specify start date, end date, and reconcilation event status parameters to determine which data to archive. Start and end dates must be in the format YYYYMMDD. For the reconciliation event parameter, you can choose Event Linked, Event Closed, or both. The Event Linked status represents events that are successfully reconciled into Oracle Identity Manager while the Event Closed' status represents events that are manually closed with Reconciliation Manager. If you choose to archive selective data, the utility disables foreign key constraints on all active reconciliation tables. The foreign key constraints will be reenabled after the archived data is deleted from the active reconciliation tables.

When archive all data from the active reconciliation tables to the archive reconciliation tables, the Reconciliation Archival utility archives all reconciliation data with a status of Event Linked or Event Closed.

The files that make up the Oracle Database version of the Reconciliation Archival utility are located in the following directory:

installServer/xellerate/db/oracle/Utilities/ReconArchival

The files that make up the SQL Server version of the Reconciliation Archival utility are located in the following directory:

installServer/xellerate/db/sqlserver/Utilities/ReconArchival

Note: Data that has been archived from the active reconciliation tables to the archive reconciliation tables will no longer be available through Oracle Identity Manager. To access this data, you must query the archive reconciliation tables in your Oracle Identity Manager database.

Preparing Oracle Database for the Reconciliation Archival Utility

Before you can use the Reconciliation Archival utility with Oracle Database, you must perform the following steps:

1. Start Oracle SQL*Plus and connect to Oracle Database as `SYS` user.
2. Create a separate table space for the archival reconciliation tables by entering the following command. Replace *DATA_DIR* with the directory where you want to

store the data file and adjust the size and other parameters as necessary for your environment.

```
CREATE TABLESPACE OIM_RECON_ARCH
    DATAFILE 'DATA_DIR\reconarch_01.dbf' SIZE 1000M REUSE
    EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
```

Note: Oracle recommends that you allocate a large UNDO tablespace when archiving large amounts of data.

3. Connect to the Oracle Database as the Oracle Identity Manager database user.
4. Enter the following command to run the Create_recon_arch_tables.sql script, which creates the archive reconciliation tables:

```
@ path/Create_recon_arch_tables.sql
```

5. Enter the following command to run the cr_recon_ddl_table.sql script, which creates a table named oim_recon_ddl. The oim_recon_ddl table is used by the Reconciliation Archival utility.

```
@ path/cr_recon_ddl_table.sql
```

6. Enter the following command to run the OIM_SP_ReconArchival.sql script, which creates a stored procedure that the Reconciliation Archival utility uses to archive and delete reconcilation data:

```
@path/OIM_SP_ReconArchival.sql
```

Preparing Microsoft SQL Server for the Reconciliation Archival Utility

Before you can use the Reconciliation Archival utility with Microsoft SQL Server, you must perform the following steps:

1. Start SQL Query Analyzer and connect to SQL Server with as a user that is a member of `sysadmin`, or who has a `dbcreator` server role or `db_owner` database role.
2. Enter the following commands. Replace *DATA_DIR* with the directory where you want to store the data file and adjust the `SIZE`, `MAXSIZE`, and `FILEGROWTH` parameters as necessary for your environment. These commands create the `OIM_ARCH_RECON` file group, which the Reconciliation Archival utility uses to store archival reconciliation data.

```
USE master
GO
ALTER DATABASE oim_database_name
ADD FILEGROUP OIM_RECON_ARCH
GO
ALTER DATABASE oim_database_name
ADD FILE
 (NAME = OIM_RECON_ARCH_01,
  FILENAME = 'DATA_DIR\RECON_ARCH_01.NDF',
  SIZE = 1000MB,
  MAXSIZE = 5000MB,
  FILEGROWTH = 25MB)
TO FILEGROUP OIM_RECON_ARCH
GO
```

3. Disconnect from SQL Server and reconnect again as the Oracle Identity Manager database user.
4. Load and execute the *path*/Create_recon_arch_tables.sql script, which creates the archive reconciliation tables.
5. Load and execute the *path*/OIM_SP_ReconArchival.sql script, which creates a stored procedure that the Reconciliation Archival utility uses to archive and delete reconcilation data.

Running the Reconciliation Archival Utility

Perform the following steps to run the Reconciliation Archival utility:

1. Ensure that the Oracle Identity Manager database is available and that no reconciliation processes are running.

   Note: Oracle recommends that you run the Reconciliation Archival utility during off-peak hours.

2. On Linux/UNIX platforms, run the following commands to set execution permission for the OIM_ReconArch.sh file and to ensure that the file is a valid Linux/UNIX text file:

   ```
   chmod 755 path/OIM_ReconArch.sh
   dos2unix path/OIM_ReconArch.sh
   ```

3. On Linux/UNIX platforms, run the *path*/**OIM_ReconArch.sh** file. On Windows platforms, run the *path***OIM_ReconArch.bat** file.
4. For Oracle Database installations, enter values for the following parameters when prompted:
   - Oracle home directory
   - Oracle Identity Manager database name or TNS string if the Oracle Identity Manager database is running on a remote machine
   - Oracle Identity Manager database user name and password

   For Microsoft SQL Server installations, enter values for the following parameters when prompted:
   - Server name where the SQL Server database is running
   - Oracle Identity Manager database name
   - Oracle Identity Manager database user name and password
5. When prompted, select one of the following options:
   - 1) Archive data from active reconciliation tables
   - 2) Delete all data from archival reconciliation tables
   - 3) Delete all data from active reconciliation tables
   - 4) Exit
6. If you selected to archive data, perform the following procedures:
   a. Select one of the following archival options:

      1) Archive selective data

2) Archive selective data by dropping and recreating indexes for faster performance

3) Archive all data

4) Exit

b. If you chose to archive selective data, enter start and end dates in the format *YYYYMMDD* when prompted.

WARNING: Be sure to enter an end date that is later than or equal to the start date or no data will be archived.

c. Select reconciliation event status for the data that you want to archive:

- Enter '1' for Closed

- Enter '2' for Linked

- Enter '3' for Closed and Linked

7. If you selected to delete data from either the archival reconciliation tables or active reconciliation tables, enter **Y** when prompted to confirm that you want to delete the data.

Output Files Generated by the Reconciliation Archival Utility

Table 8–1 describes the output files that are generated by the Reconciliation Archival utility.

Table 8–1 Output Files Generated by the Reconciliation Archival Utility

File	Description
Err_DB_Conn_*timestamp*.log	Generated when the utility is unable to connect to the database with the provided credentials
Err_Arch_Recon_*timestamp*.log	Generated when the archival or deletion processes fail
Arch_Recon_*timestamp*.log	Generated when the archival or deletion processes succeed

Index

T

W