Oracle® Real-Time Decisions Base Application

Decision Management Installation and Configuration Guide

Release 3.1.1

E28941-01

April 2012

ORACLE®

Oracle Real-Time Decisions Base Application Decision Management Installation and Configuration Guide, Release 3.1.1

E28941-01

Copyright © 2010, 2012, Oracle and/or its affiliates. All rights reserved.

Primary Author: Oracle Corporation

Contributors: Oracle Real-Time Decisions development, product management, and quality assurance teams.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface

This document describes the installation and configuration of Oracle Real-Time Decisions (Oracle RTD) Decision Management applications.

Audience

This document is intended for the following Oracle RTD users:

- Technical users configuring Oracle RTD Decision Management applications using Decision Designer
- Administrators

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Real-Time Decisions platform version 3.0 documentation set and the Oracle Real-Time Decisions Base Application Release 3.1.1 documentation set:

- *Oracle Real-Time Decisions Installation and Administration Guide*
- *Oracle Real-Time Decisions Platform Developer's Guide*
- *Oracle Real-Time Decisions Decision Center User's Guide*
- *Oracle Real-Time Decisions Release Notes*
- *Oracle Real-Time Decisions Base Application Installation and Reference Guide*
- *Oracle Real-Time Decisions Base Application Decision Management Applications User's Guide*

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.

New Features for Oracle RTD Base Application Developers and Users

This preface describes the new features of Oracle RTD Base Application Release 3.1.1.

New Features for Oracle RTD Base Application Release 3.1.1

New features of Oracle RTD Base Application Release 3.1.1, all of which relate to Oracle RTD Decision Management applications, include:

- Ability to Edit Projects and Their Attributes
- Ability to Specify Descriptions for Perspectives
- Support for Many-To-Many Relationships
- Folders and Multiple Relationships in Perspective Trees
- Configurable Pages Layout and Sequence
- Default Values for Choice and Project Attributes
- Image Widget to Visualize Associated Content
- Support for Custom Widgets
- Customization Framework
- Direct Access to Main Repository and Project Audit Trail Information
- Support for WebSphere Application Server
- Applications Can Be Configured with Optional Decision Center Reporting
- Multiple Oracle RTD Decision Management Applications Deployable to Same Domain

Ability to Edit Projects and Their Attributes

You can specify attributes for projects, and set project attribute values during project creation and editing. For instance, the Type attribute has been added to the released Oracle RTD Decision Management application, RTD for Marketing Optimization, which enables you to select Emergency Changes, Incremental Changes, or Structural Changes.

A Details icon has been added to the list of work icons in an Oracle RTD Decision Management application, to enable easy access to the project detail view and edit screens.

See Section 2.2.2.3, "Choice Group, Project, and Relationship-Types XML Files" and Section 2.2.5.1, "Adding an Attribute to a Choice Group or Project" for more information.

Ability to Specify Descriptions for Perspectives

Descriptions for perspectives, when specified, will appear as headings in perspective panels in the Oracle RTD Decision Management application user interface. They will also appear for mouse hovers over perspective names when users view and select perspectives in the dropdown perspective list.

See Section 2.2.2.2, "Perspectives XML File" for more information.

Support for Many-To-Many Relationships

You can now specify many-to-many relationships between choice groups. Oracle RTD Decision Management supports required and non-required relationships.

Non-required relationships can have two choice groups, where choices of one type do not have to be associated with any choices of the other type. The Oracle RTD Decision Management application RTD for Marketing Optimization has been enhanced by the addition of the choice group Tag, which has a many-to-many relationship with the choice group Creative. With this relationship, each creative can have multiple tags, and each tag can be associated with many creatives. Creatives do not have to have a tag, and tags do not have to be associated with any creative.

An example of the required relationship type would be between campaigns and products, where each campaign must have at least one product, but a product does not have to be associated with a campaign.

As with other kinds of relationship, many-to-many relationships can be configured to propagate events and rules.

See Section 2.2.2.3, "Choice Group, Project, and Relationship-Types XML Files" for more information.

Folders and Multiple Relationships in Perspective Trees

In the previous release, you could display hierarchical relationships between choice groups as indented nodes in an Oracle RTD Decision Management application perspective panel tree. In this release, you can now define multiple relationships at each node of the hierarchy tree. You can display multiple relationships under a node with or without the use of a folder to group each set of related choices.

For example, marketing collateral documents could have many authors and refer to several competitors. In an Oracle RTD Decision Management application perspective panel, with the extra folders showing, this could appear as:

```
Central Region Release Collateral
  Has Author
    (and under this, many individual authors)
  Has Competitor
    (and under this, many different competitors)
```

See Section 2.2.2.2, "Perspectives XML File" for more information.

Configurable Pages Layout and Sequence

In the previous release, while the pages which displayed for choice creation, editing, or viewing varied according to the choice attributes, the pages (and sequence of pages) were generated automatically and were not configurable.

In this release, the content, layout, and sequence of the pages is explicitly defined through metadata associated with the choices and projects.This also enables user-specific choice lifecycle management as pages also have role-based permissions.

See Section 2.2.2.4, "User Interface Pages" for more information.

Default Values for Choice and Project Attributes

You can specify default values for choice and project attributes, which appear on Oracle RTD Decision Management application pages as you create new choices and projects.

The type of values supported are string, number, date. They can be set at design time in metadata, or at run time by having a Groovy expression in metadata.

This feature is complementary to the fact that choice attributes can have default values defined in the associated Inline Service.

See Section 2.2.2.3, "Choice Group, Project, and Relationship-Types XML Files" for more information.

Image Widget to Visualize Associated Content

For enhanced visual displays of choices and projects in Oracle RTD Decision Management applications, you can specify image widgets for choice or project attributes. This enables the images to appear when the choice or project is opened in view mode. For example, when viewing a creative in the Oracle RTD Decision Management application RTD for Marketing Optimization, you can visualize the image associated with the content in the CMS repository which will be presented to the end user from the CMS repository. This provides a consistent end-user view across all channels.

See Section 2.2.2.3, "Choice Group, Project, and Relationship-Types XML Files" for more information.

Support for Custom Widgets

You can create your own widgets and associate them with choice or project attributes. All the Oracle RTD Decision Management widgets now use this framework. By using this framework, custom widgets will be safeguarded during Oracle RTD Decision Management upgrades.

Customization Framework

You can use Decision Designer to customize the behavior of Oracle RTD Decision Management applications by adding your own logic. The two ways to achieve customization are through extensibility of the ADF Framework and through the Oracle RTD Decision Management templates.

Direct Access to Main Repository and Project Audit Trail Information

An Audit Trail icon has been added to the Main Repository/Project context area in the Oracle RTD Decision Management application, to enable easier access to audit trail information for the environment in which you are working.

If you have selected to view the Main Repository only, clicking Audit Trail will display audit trail information for objects in the Main Repository. If you are working in a project, clicking Audit Trail will display audit trail information for objects in the current project.

Support for WebSphere Application Server

In this release, you can choose whether to deploy and run production Oracle RTD Decision Management applications under WebSphere Application Server or WebLogic.

See Section 1.1, "Installation Overview" for more information, and also sections specific to WebSphere or WebLogic in Chapter 1, "Installing Oracle RTD Decision Management."

Applications Can Be Configured with Optional Decision Center Reporting

In this release, you can choose whether to enable Decision Center reporting from the Oracle RTD Decision Management application. With Decision Center reporting enabled, Performance and Analysis tabs appear in the main choice details area of the Oracle RTD Decision Management application user interface. If Decision Center reporting is not enabled, these tabs do not appear.

See Section 2.2.2.1, "Config XML File" for more information.

Decision Center reporting is only supported with homogenous application server deployments. Additionally, SSO must be configured between Oracle RTD Decision Management and Decision Center (for details, see Section 1.6.7, "Using SSO with Oracle RTD Decision Management Applications on WebLogic" and Section 1.6.8, "Using SSO with Oracle RTD Decision Management Applications on WebSphere.")

Heterogeneous topologies are supported, but without Decision Center reports integration within Oracle RTD Decision Management.

Multiple Oracle RTD Decision Management Applications Deployable to Same Domain

In this release, you can now deploy many Oracle RTD Decision Management applications to the same domain.

See Section 2.2.2.1, "Config XML File" for more information.

1

Installing Oracle RTD Decision Management

Terminology: The term "reference implementation" is used in this chapter to refer to the specific Oracle RTD Decision Management application Oracle RTD for Marketing Optimization (also referred to as the RTD for Marketing Optimization application) released with Oracle RTD Base Application.

This chapter contains the following topics:

1.1 Installation Overview

The steps that lead to the deployment of a Oracle RTD Decision Management application are as follows:

- Using Oracle JDeveloper, a developer configures the Oracle RTD Decision Management application and deploys it to the Weblogic Server that comes with JDeveloper.

  The application server for this stage *must* be WebLogic.

- After the application has been properly configured, the developer creates an Oracle RTD Decision Management application EAR file, preparing for the file to be deployed to a target production (or test) environment.

  The target environment may use either WebLogic or WebSphere for its application server, and the preparation steps - performed in the development environment - are slightly different for the two different target application servers.

- The domain administrator deploys the EAR file to either a WebSphere server or to a remote Oracle WebLogic Server using Enterprise Manager to be used in the production (or test) environment.

Summarizing, there are two environments:

- Development environment - the environment for the configuration of Oracle RTD Decision Management *(must use WebLogic)*
- Production environment - the environment for the use of the configured Oracle RTD Decision Management application either for testing or production *(must use either WebLogic or WebSphere)*

Note: When following references to the *Oracle Real-Time Decisions Installation and Administration Guide*, please first follow the installation instructions bundled with the specific Oracle RTD platform version required for Oracle RTD Decision Management (as specified in *Oracle Real-Time Decisions Applications System Requirements and Supported Platforms*).

1.2 Installing Oracle RTD Decision Management for Development

This section contains the following topics:

1.2.1 Preparing for Installation

The outline steps to perform in preparing to install Oracle RTD Decision Management are as follows *(if details are required, use the appropriate product documentation)*:

1. Install Oracle Database.
2. Install Oracle JDeveloper 11g Rel 1.
3. Install Oracle RTD for WebLogic, then create and initialize the Oracle RTD database.
4. Create a WebLogic domain with an Administration Server only.
5. Configure Oracle RTD to run on the Administration Server.

The rest of this section contains the following topics:

1.2.1.1 Preparing to Use Oracle JDeveloper

After installing Oracle JDeveloper 11g Rel 1, perform the following steps:

1. In JDeveloper:
   - Select Tools, then Preferences.
   - Select the Environment tab and set the Encoding to UTF-8.
   - Click OK.
2. Create the following environment variables:
   - ANT_HOME=`<Oracle Middleware Install dir>\jdeveloper\ant`
   - JAVA_HOME=`<Oracle Middleware Install dir>\<jdk_version>`

Add %ANT_HOME%\bin and %JAVA_HOME%\bin to your PATH.

3. Create a file called clm-build.properties in your home directory (on Windows, this folder can be reached using the variable %HOMEPATH%). This file should contain:
   - jdeveloper.home=*<Oracle Middleware Install dir>*

     For example, C:/Oracle/Middleware.

     Note: Use the forward slash character "/" in the file, even on Windows.

1.2.1.2 Setting Up Oracle RTD and the Oracle RTD Database

To set up Oracle RTD and the Oracle RTD database, install Oracle RTD for WebLogic, then create and initialize the Oracle RTD database, as described in Sections 2.1 and 2.2 of *Oracle Real-Time Decisions Installation and Administration Guide*.

Note: In addition to the instructions in the specified sections, you must make the following extra provision for the Oracle RTD database SDDB:

- Before you can initialize the database either by using SDDBTool or by running the database initialization command line scripts, you must have the system privilege CREATE VIEW.

1.2.1.3 Creating the WebLogic Domain

The WebLogic domain to create for the development environment will have the following properties:

- The domain will be configured to support Oracle JRF
- The domain will have only one Administration Server

To create the development WebLogic domain, perform the following steps:

1. Start the Configuration Wizard.

   On Windows, this is available via Start Menu: Programs -> Oracle Fusion Middleware *<11g_version>* -> WebLogic Server 11gR1 -> Tools -> Configuration Wizard
2. In the Welcome window, select **Create a new WebLogic domain**, then click Next.
3. In the Select Domain Source window:
   - Select **Generate a domain configured automatically to support the following products**.
   - Select the **Oracle JRF - 11.1.1.0 [oracle_common]** option.
   - Click Next.
4. In the Specify Domain Name and Location, specify a name and location for your domain, then click Next.

> **Note:** You can specify any name for the domain. The generic name name ***<RTDCLM_Dev_Domain>*** will be used to refer to the development environment domain name in all the setup steps of this and other sections.

5. In the Configure Administrator User Name and Password window, specify a user name and password for the administrator account, then click Next.
6. In the Configure Server Start Mode and JDK window, keep the default values in the **WebLogic Domain Startup Mode** and **Available JDKs** areas, then click Next.
7. In the Select Optional Configuration window, for the development environment select only **Administration Server**, then click Next.
8. In the Configure the Administration Server window:
   - Keep the default values for Name and Listen address.
   - For Listen port, either keep the default value of 7001 or enter the port number that you require for the Administration Server.
   - Click Next.
9. Review the domain configuration details in the Configuration Summary window, then click Create.

1.2.1.4 Configuring Oracle RTD

This section describes the configuring of Oracle RTD to run on the Administration Server.

> **Note:** The rationale for running Oracle RTD on the Administration Server in the development environment is to use less memory. However this should never be used in a production environment.

With reference to the instructions in chapter 5 of the *Oracle Real-Time Decisions Installation and Administration Guide*, perform the following steps:

1. If you are running Oracle RTD in a Managed Server, perform the instructions in Section 5.4.

   If you are running Oracle RTD in the Administration Server instead of a Managed Server, replace Section 5.4 with these instructions:

   For **Windows**: In the bin subfolder of your domain, edit `startWebLogic.cmd` and add the following entries at the top of the file:

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.S
DEMFRegistry
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SA
XParserFactoryImpl
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote=true
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.port=12345
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Dcom.sun.management.jmxremote.authenticate=false
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.ssl=false
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dweblogic.wsee.skip.async.response=true
```

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Djava.net.preferIPv4Stack=true
```

For **Unix**: In the bin subfolder of your domain, edit `startWebLogic.sh` and add at the top of the file:

```
JAVA_OPTIONS="${JAVA_OPTIONS}
-Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.S
DEMFRegistry"
JAVA_OPTIONS="${JAVA_OPTIONS}
-Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SA
XParserFactoryImpl"
JAVA_OPTIONS="${JAVA_OPTIONS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTIONS="${JAVA_OPTIONS} -Dcom.sun.management.jmxremote.port=12345"
JAVA_OPTIONS="${JAVA_OPTIONS}
-Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTIONS="${JAVA_OPTIONS} -Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTIONS="${JAVA_OPTIONS} -Dweblogic.wsee.skip.async.response=true"
JAVA_OPTIONS="${JAVA_OPTIONS} -Djava.net.preferIPv4Stack=true"
```

2. Do *not* perform the steps in Section 5.5.
3. If you are running Oracle RTD in a Managed Server, perform the instructions as they appear from Section 5.6 to the end of Chapter 5.

   If you are running Oracle RTD in the Administration Server instead of a Managed Server, perform the operations from Section 5.7 to the end of Chapter 5 on the Administration Server instead of the Managed Server.

1.2.2 Installing Oracle RTD Decision Management

After you have created the WebLogic domain and configured Oracle RTD to run on the Administration Server, you are ready to install Oracle RTD Decision Management.

> **Note:** This section describes how to install Oracle RTD Decision Management and the specific RTD for Marketing Optimization application, as released with Oracle RTD Base Application. For setting up other Oracle RTD Decision Management applications, see Section 2.2, "Configuring Oracle RTD Decision Management."

This section contains the following topics:

- Section 1.2.2.1, "Unzipping the Oracle RTD Decision Management Product"
- Section 1.2.2.2, "Oracle RTD Decision Management Database Creation"
- Section 1.2.2.3, "Oracle RTD Decision Management Data Source Setup"
- Section 1.2.2.4, "Oracle RTD Decision Management Data Source Access by Oracle RTD Setup"
- Section 1.2.2.5, "Inline Service Deployment"
- Section 1.2.2.6, "Storing Credentials to Enable Web Service Calls"
- Section 1.2.2.7, "Oracle RTD Decision Management Application Deployment"

1.2.2.1 Unzipping the Oracle RTD Decision Management Product

Unzip `RTD_Designer.zip` from the Oracle RTD Base Application distribution into the directory of your choice.

Note: Some zip extraction utilities impose a limit on the combined length of the target directory path and the full directory+file name in the zip file. Oracle recommends that select your target directory with this in mind when you unzip `RTD_Designer.zip`.

1.2.2.2 Oracle RTD Decision Management Database Creation

Note: For running the SQL commands and SQL scripts specified in this section, use SQL*Plus or an equivalent tool that can run on your database, such as the Database Navigator of Oracle JDeveloper.

To create an Oracle RTD Decision Management database, perform the following steps:

1. Create a database user for the Oracle RTD Decision Management schema.

   ```
   CREATE USER <username> IDENTIFIED BY <password> DEFAULT
   TABLESPACE users TEMPORARY TABLESPACE temp QUOTA UNLIMITED ON
   users;

   GRANT CREATE VIEW, CONNECT, RESOURCE TO <username>;
   ```

   Note: If you are going to use non 7-bit ASCII characters in choice group id or choice attribute ids, you must set the environment variable NLS_LANG to .UTF8 before running the sql scripts in the steps that follow.

2. Create the schema, by performing the following steps:

   a. *Preliminary step if you want to use Oracle JDeveloper:*

      Open Oracle JDeveloper, open Database Navigator, then connect as the user that you just created.

   b. Go to the subfolder **clm\Database\sql** of where you unzipped `RTD_Designer.zip`, then run the file `load core.sql`.

      Note: The file `load core.sql` contains generic commands, required for all Oracle RTD Decision Management applications. The commands in `load ils.sql`, as released, are for the RTD for Marketing Optimization application.

   c. Go to the subfolder **clm\Database\sql\ils** of where you unzipped `RTD_Designer.zip`, then run the file `load ils.sql`.

      Make sure you have committed your changes.

      Note: The following step is specific to the installation of the application RTD for Marketing Optimization, as released with Oracle RTD Base Application. The step seeds the database with some sample data.

3. Load the sample seed data into the Oracle RTD Decision Management database for the reference implementation application, by performing the following steps:

   - Run `clm\Build\metadata\ref\sql\insert ils data.sql` first to create some channels, placements, slots and slot types.
   - Run `clm\Build\metadata\ref\sql\insert marketing data.sql` to create some campaigns, offers and creatives.

   You may notice the `drop core.sql` and `drop ils.sql` files. These can be used to drop the schema. Call `drop ils.sql` first, then `drop core.sql`.

1.2.2.3 Oracle RTD Decision Management Data Source Setup

To set up an Oracle RTD Decision Management data source, perform the following steps:

1. Start the Administration Server.

   On Windows, this is available via the Start Menu: Programs -> Oracle Fusion Middleware *<11g_version>* -> User Projects -> *<RTDCLM_Dev_Domain>* -> Start Admin Server for Weblogic Server Domain

2. Open the Admin Server Console.

   On Windows, this is available via the Start Menu: Programs -> Oracle Fusion Middleware *<11g_version>* -> User Projects -> *<RTDCLM_Dev_Domain>* -> Admin Server Console.

3. Log in with the administrator username and password, which was specified during domain creation (see Section 1.2.1.3, "Creating the WebLogic Domain.").

4. Navigate the path Services -> Data Sources -> New -> Generic Data Source, then enter:

   - Name: **DM_DS** or a similar value
   - JNDI Name: **CLMDS**
   - Database Type: **Oracle**

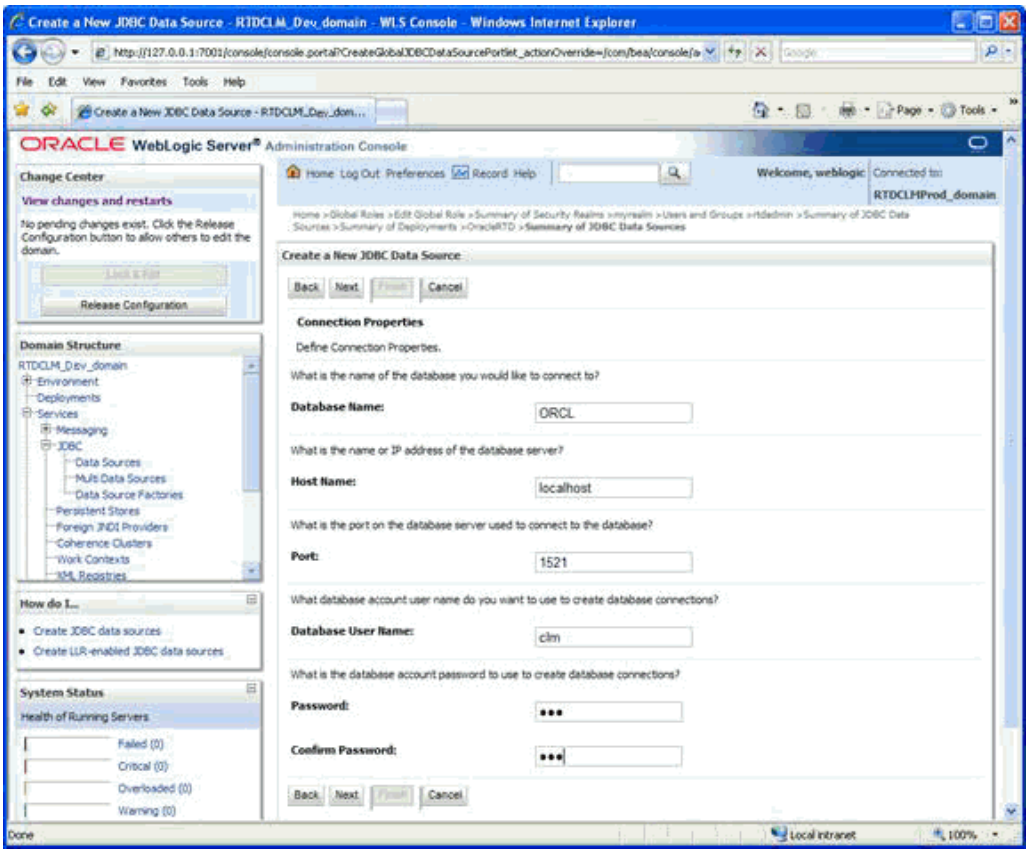
   Click Next.

5. For the Database Driver, select **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later**, then click Next.

6. In the Transaction Options window, deselect **Supports Global Transactions**, then click Next.

7. For the Connection Properties:

   - Specify the database settings that match your configuration, for example:



Ensure that the Database User name and Password match the values that you set up in step 1 of Section 1.4.3.2, "Oracle RTD Decision Management Database Creation."

- Click Next.

8. Leave all the settings already filled, except enter CHOICE for the Test Table Name, and click Test Configuration. Then click Next.
9. Select AdminServer as a target, then click Finish.

1.2.2.4 Oracle RTD Decision Management Data Source Access by Oracle RTD Setup

Section 8.3.3 of *Oracle Real-Time Decisions Installation and Administration Guide* describes the general steps to add a data source to Oracle RTD. Follow the steps in that section, and, specifically for the Oracle RTD Decision Management data source, add the following in `web.xml` for `rtis.war`:

```
<resource-ref id="CLM_RTIS">
    <res-ref-name>CLMDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

and add the following in `web.xml` for `soap.war`:

```
<resource-ref id="CLM_Axis">
    <res-ref-name>CLMDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
```

```
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

1.2.2.5 Inline Service Deployment

To deploy the Oracle RTD Decision Management application, you must have a specific Inline Service running in the Oracle RTD instance on the same server as Oracle RTD Decision Management. You need to deploy that Inline Service to the Oracle RTD server using Decision Studio, as described in *Oracle Real-Time Decisions Platform Developer's Guide*.

By convention, the Inline Service is located in a sub-folder called `service` of the application module folder. For the reference implementation, that folder is **clm\Build\metadata\ref\service**.

1.2.2.6 Storing Credentials to Enable Web Service Calls

Oracle RTD Decision Management makes web service calls from the Decision Management server side to Oracle RTD. Credentials for these calls are stored in the WebLogic credential store. In summary form, the credential setup process is as follows:

- First, you create a user that will be used to authenticate the web service calls
- Then, you store the credentials of this user in the WebLogic credential store, so that the Decision Management application can retrieve them

To create the user, perform the following steps:

1. Open the Admin Server Console.

   On Windows, this is available via the Start Menu: Programs -> Oracle Fusion Middleware *<11g_version>* -> User Projects -> *<RTDCLM_Dev_Domain>* -> Admin Server Console.
2. Log in with the administrator username and password.
3. Select Security Realms and select the security realm you are using (myrealm by default).
4. Select the Users and Groups tab, and click New to create a new user.
5. Specify a username and password (generic names *<clmclient_user>* and *<clmclient_password>*).
6. Click OK.
7. Select the user that you just created and select the Groups tab.
8. Select the RTDDCUserGroup group, shuttle it to the right.
9. Click Save:

To store the username and password in the WebLogic credential store, perform the following steps:

1. In a command window, go to the directory **<*middleware_home*>\oracle_common\common\bin**, and run `wlst.cmd` on Windows or `wlst.sh` on Unix.

   Note: **<*middleware_home*>\wlserver_10.3\common\bin** includes similarly named command files - do *not* execute those.

2. Using the values appropriate to your environment (specified during domain creation), type the following command:

   ```
   connect('<admin_user>', '<admin_password>', '<server_name>:<port>')
   ```

3. Using the same username and password that you created previously in this section, enter:

   ```
   updateCred(map="oracle.rtd.clm", key="clm.client",
   user="<clmclient_user>", password="<clmclient_password>")
   ```

4. Verify that this worked, by entering:

   ```
   listCred(map="oracle.rtd.clm", key="clm.client")
   ```

5. Complete the process by entering:

   ```
   exit()
   ```

Note: The user name and password are required as part of the SSO setup for Decision Management, as described in Section 1.6.7.4, "Creating Groups and Users in LDAP Server."

1.2.2.7 Oracle RTD Decision Management Application Deployment

Deploy the Oracle RTD Decision Management application in JDeveloper, by performing the following steps:

1. Open JDeveloper from Start Menu, Programs -> Oracle Fusion Middleware *<11g_version>* -> JDeveloper Studio *<11g_version>*.

   If prompted, select Default Role.

2. Click Open Application... and select **clm.jws** in the clm subfolder of where you unzipped `RTD_Designer.zip`.

3. Select Application -> Deploy -> clm_application1... and, in the Deployment Action window, select **Deploy to Application Server**.

4. At top right of the Select Server window, click the green plus button.

   The Create Application Server Connection process that opens up has five steps:

   - Name and Type: Select AdminServer and WebLogic 10.3.
   - Authentication: Specify the administrator username and password.
   - Configuration: Specify the Weblogic domain *<RTDCLM_Dev_Domain>*, and the Hostname and port values that apply to your environment.
   - Test: Click Test Connection.
   - Finish: Click Finish.

5. In the Select Server window, select the newly created AdminServer connection, and click Next.

6. In the WebLogic options window, leave the options as they are, and click Finish.

   You should see this in the deployment tab:

   - Application Deployed Successfully.
   - Elapsed time for deployment: [...]
   - ---- Deployment finished. ---

[Next time you can deploy the application by choosing the menu item Application -> Deploy -> clm_application1 to AdminServer.]

The Oracle RTD Decision Management application is now deployed and accessible at the URL: `http://<server>:<port>/dm`

1.3 Preparing Your Development Application for Production

After you have finished configuring Oracle RTD Decision Management for your needs, you are ready to generate the clm.ear file to provide it to the domain administrator so it can be installed in your production or test environment. The instructions for this subsequent installation are application server specific and appear in later sections of this manual.

> **Note:** The preparation steps in this section are largely the same whether your target production (or test) environment uses WebLogic or WebSphere. Unless marked as being specific to a particular target application server, the instructions in this section apply to both.

To generate the clm.ear file, first review the Oracle RTD Decision Management configuration:

- (For WebLogic production and test systems)

  In **clm\Build\metadata\<*metadata_module*>\security.xml** (and any other xml files in the same directory **clm\Build\metadata\<*metadata_module*>**), remove any <users> or <enterprise-roles> as the domain administrator will map Oracle RTD Decision Management application roles to your enterprise users and groups using Enterprise Manager (see Section 1.6.4, "Mapping Users and Groups to Intermediary Roles in Production").

- (For WebSphere production and test systems)

  In **clm\Build\metadata\<*metadata_module*>\security.xml** (and any other xml files in the same directory **clm\Build\metadata\<*metadata_module*>**), remove any <users> and create <enterprise-role> for each <application-role>.For example if you have:

```
<application-roles>
  <application-role>
      <name>CLMAdministrator</name>
...
  </application-role>
...
</application-roles>
```

  then add:

```
<enterprise-roles>
...
  <enterprise-role>
        <application-roles>CLMAdministrator</application-roles>
        <name>CLMAdministratorGroup</name>
  </enterprise-role>
...
</enterprise-roles>
```

The domain administrator will map these enterprise roles to its own enterprise users and groups. See Section 1.6.4, "Mapping Users and Groups to Intermediary Roles in Production."

- Review **clm\Build\metadata\<*metadata_module*>\config\config.xml**.

  You will probably have to change the **workbenchService url** port from the development-site port number to the port where your production managed server will be running.

  For example, for WebLogic environments, from the development-site Administration Server port 7001 to the production-site Managed Server port 7003, and for WebSphere servers, the default port is 9080.

  Note the JNDI name for your data source.

  This is the name identified by <datasourceJNDIName>, CLMDS by default. You will need this when setting up your production system.

If you have made changes to the configuration, go to the **clm\Build** directory and run the command **`ant generate`**:

Then run the command **`ant ear`**:

The `clm.ear` file is now located in the folder **clm\deploy** and is ready to be used in production.

You need to provide `load core.sql` and `load ils.sql` to set up the production database. These are located in folder **clm\Database\sql** and **clm\Database\sql\ils** respectively.

You also need to provide the Inline Service. It should be located in the folder **clm\Build\metadata\<*metadata_module*>\service**.

1.4 Installing Oracle RTD Decision Management for Production on WebLogic

This section contains the following topics:

- Section 1.4.1, "Preparing for Installation"
- Section 1.4.2, "Production WebLogic Domain Creation"
- Section 1.4.3, "Installing Oracle RTD Decision Management"

1.4.1 Preparing for Installation

The steps to perform in preparing to install Oracle RTD Decision Management are as follows:

1. Install Oracle Database.
2. Install Oracle WebLogic Server 11gR1.

   Oracle Coherence and Oracle Enterprise Pack for Eclipse do not need to be installed and can be deselected during install.
3. Install Application Development Runtime.
4. Configure the Oracle RTD database following chapter 2 of *Oracle Real-Time Decisions Installation and Administration Guide*.

Note: In addition to the instructions in the specified chapter, you must make the following extra provision for the Oracle RTD database SDDB:

- Before you can initialize the database either by using SDDBTool or by running the database initialization command line scripts, you must have the system privilege CREATE VIEW.

5. Create a WebLogic domain with a Managed Server, as explained in Section 1.4.2, "Production WebLogic Domain Creation."
6. Modify `nodemanager.properties` in `<middleware_home>/wlserver_10.3/common/nodemanager` and set **StartScriptEnabled** and **StopScriptEnabled** to true (you have to start Node Manager once for this file to exist).
7. Configure Oracle RTD to run on the Managed Server following chapter 5 (but not sections 5.1, 5.3, nor 5.5) of *Oracle Real-Time Decisions Installation and Administration Guide*.

1.4.2 Production WebLogic Domain Creation

To create a production WebLogic domain, perform the following steps:

1. Start the Fusion Middleware Configuration Wizard.

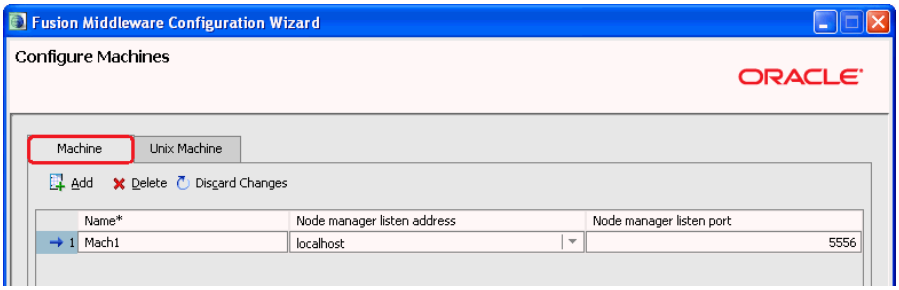
   On Windows, this is available via Start Menu: Programs -> Oracle Application Developer 11g -> Configure Application Server.
2. In the Welcome window, select **Create a new WebLogic domain**, then click Next.
3. In the Select Domain Source window:
   - Select **Generate a domain configured automatically to support the following products**.
   - Select the **Oracle Enterprise Manager - 11.1.1.0 [oracle_common]** and **Oracle JRF - 11.1.1.0 [oracle_common]** options.
   - Click Next.
4. In the Specify Domain Name and Location, specify a name and location for your domain, then click Next.

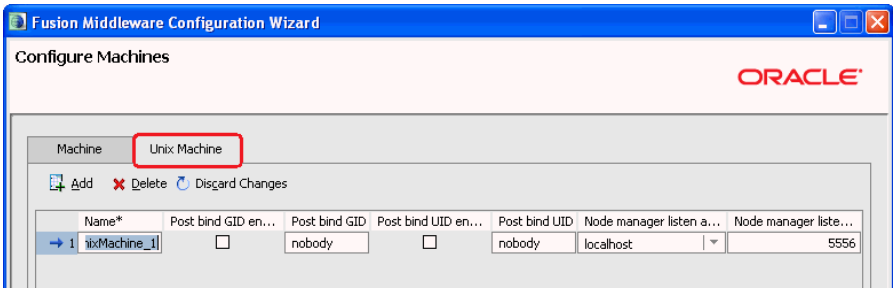
   Note: You can specify any name for the domain. The generic name name ***<RTDCLM_Prod_Domain>*** will be used to refer to the production environment domain name in all the setup steps of this and other sections.

5. In the Configure Administrator User Name and Password window, specify a user name and password for the administrator account, then click Next.
6. In the Configure Server Start Mode and JDK window, select **Production Mode**, then click Next.
7. In the Select Optional Configuration window, select the following options, then click Next.
   - **Administration Server**

- **Managed Servers, Clusters and Machines**
- **Deployment and Services**

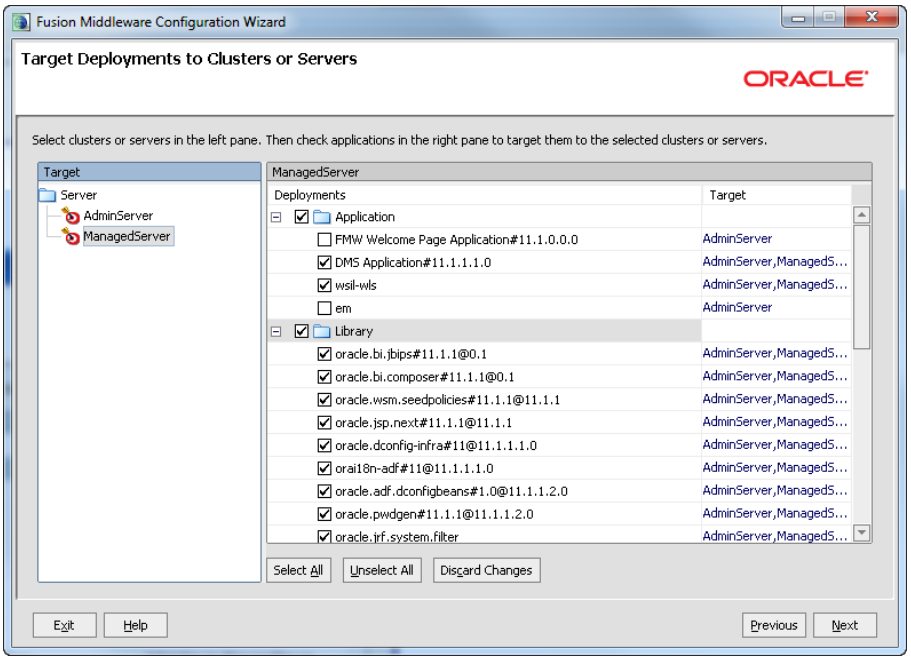
8. In the Configure the Administration Server window, optionally change the values as required for your production environment, then click Next
9. In the Configure Managed Servers window, add a managed server specifying a name of your choice, with address and port information appropriate to your environment, then click Next.
10. In the Configure Clusters window, click Next.
11. In the Configure Machines window:
    - Select the Machine tab (for Windows) or the UNIX Machine tab (for UNIX)





    - Add a machine specifying a name of your choice, and with address and port information appropriate to your environment

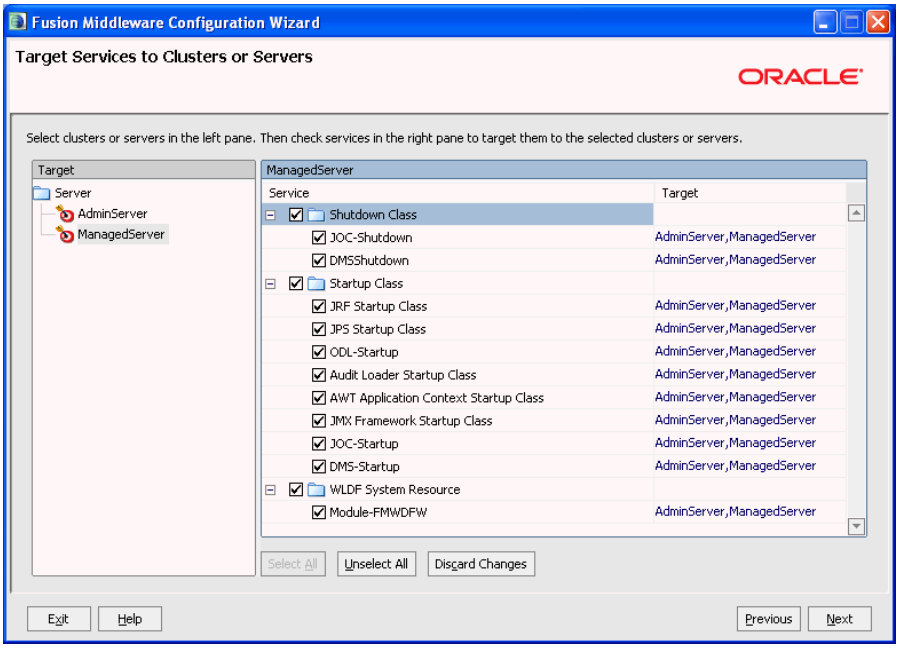
      Note the extra "Post bind" fields on the Unix Machine tab. Click Help in the Configure Machines window for further information on these fields, to determine their relevance for your environment.
    - Click Next
12. In the Assign Servers to Machines window, assign your managed server to the machine that you just specified, then click Next.
13. In the Target Deployments to Clusters or Servers window:
    - Select **ManagedServer** as the Target
    - Select the applications **DMS Application#11.1.1.1.0** and **wsil-wls**
    - Select **Library** (which automatically selects all the libraries)



- Click Next

14. In the Target Services to Clusters or Servers window:

    - Specify that all the services should be targeted to the Managed Server.



- Click Next.

15. Review the domain configuration details in the Configuration Summary window, then click Create.

Note: If you have performed these domain creation steps in sequence from Section 1.4.1, "Preparing for Installation," continue those steps at step 6.

Other Configurations

For your own installation, you may have a different configuration based on your topology. For example:

- You can have the Administration Server on a different machine
- You can have multiple Managed Servers
- You can use a cluster
- You can enable SSL (see Section 1.6.5, "Using SSL with Oracle RTD Decision Management Applications on WebLogic")

1.4.3 Installing Oracle RTD Decision Management

After you have created the WebLogic domain and configured Oracle RTD to run on the Managed Server, you are ready to install Oracle RTD Decision Management.

This section contains the following topics:

- Section 1.4.3.1, "Files Required from Development Environment"
- Section 1.4.3.2, "Oracle RTD Decision Management Database Creation"
- Section 1.4.3.3, "Oracle RTD Decision Management Data Source Setup"
- Section 1.4.3.4, "Oracle RTD Decision Management Data Source Access by Oracle RTD Setup"
- Section 1.4.3.5, "Storing Credentials to Enable Web Service Calls"
- Section 1.4.3.6, "Oracle RTD Decision Management Application Deployment"
- Section 1.4.3.7, "Inline Service Deployment"

1.4.3.1 Files Required from Development Environment

The Oracle RTD Decision Management configuration performed in the development environment is the source of the following files, required for the production environment:

- clm.ear
- load core.sql
- load ils.sql
- The Inline Service that is part of this application (see Section 1.2.2.5, "Inline Service Deployment")

Note: The .sql files are generated when you run **ant generate** (and before you run **ant ear**).

1.4.3.2 Oracle RTD Decision Management Database Creation

> **Note:** For running the SQL commands and SQL scripts specified in this section, use SQL*Plus or an equivalent tool that can run on your database.

To create an Oracle RTD Decision Management database, perform the following steps:

1. Create a database user for the Oracle RTD Decision Management schema.

   ```
   CREATE USER <username> IDENTIFIED BY <password> DEFAULT
   TABLESPACE users TEMPORARY TABLESPACE temp QUOTA UNLIMITED ON
   users;

   GRANT CREATE VIEW, CONNECT, RESOURCE TO <username>;
   ```

   > **Note:** If you are going to use non 7-bit ASCII characters in choice group id or choice attribute ids, you must set the environment variable NLS_LANG to .UTF8 before running the sql scripts in the steps that follow.

2. Create the schema, by performing the following steps:

   - Go to the subfolder **clm\Database\sql** of where you unzipped `RTD_Designer.zip`, then run the file `load core.sql`.

     > **Note:** The file `load core.sql` contains generic commands, required for all Oracle RTD Decision Management applications. The commands in `load ils.sql`, as released, are for the RTD for Marketing Optimization application.

   - Go to the subfolder **clm\Database\sql\ils** of where you unzipped `RTD_Designer.zip`, then run the file load ils.sql.

     Make sure you have committed your changes.

1.4.3.3 Oracle RTD Decision Management Data Source Setup

To set up an Oracle RTD Decision Management data source, perform the following steps:

1. Start the Administration Server.

   On Windows, this is available via the Start Menu: Programs -> Oracle WebLogic -> User Projects -> *<RTDCLM_Prod_Domain>* -> Start Admin Server for Weblogic Server Domain

2. Enter the administrator username and password, which was specified during domain creation (see Section 1.2.1.3, "Creating the WebLogic Domain").

3. Open the Admin Server Console.

   On Windows, this is available via the Start Menu: Programs -> Oracle WebLogic -> User Projects -> *<RTDCLM_Prod_Domain>* -> Admin Server Console.

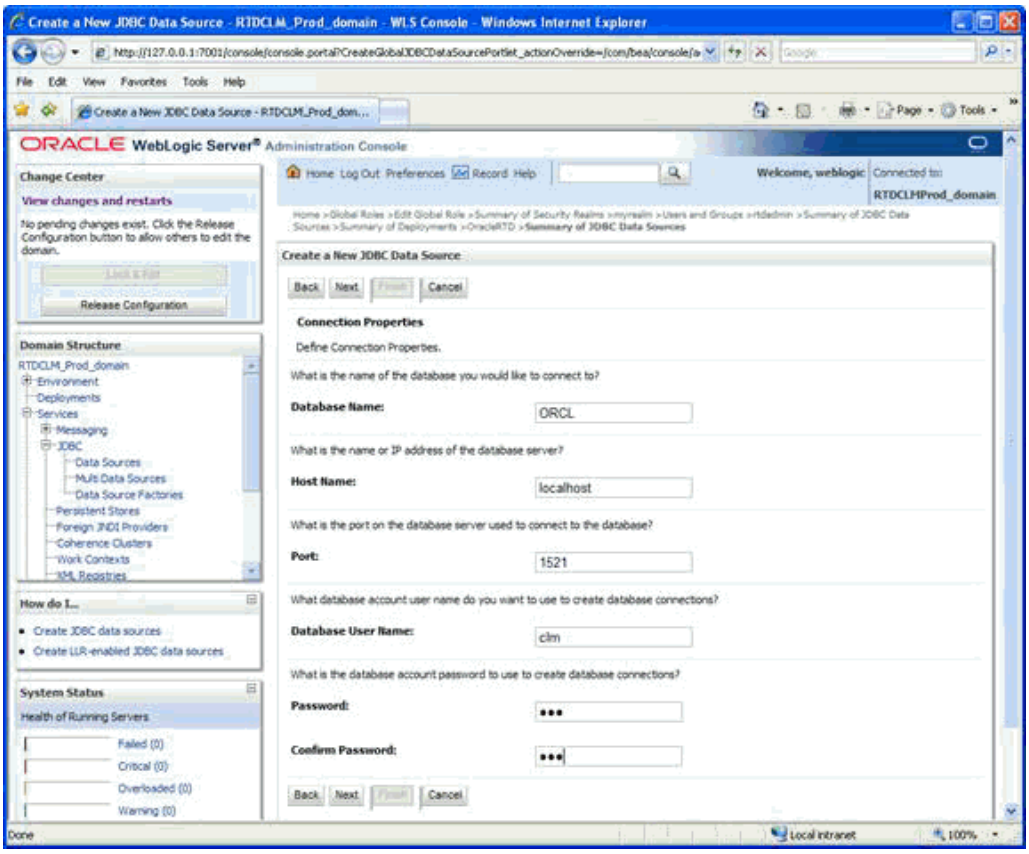
4. Log in with the administrator username and password.

5. Click **Lock & Edit** to make change.

6. Navigate the path Services -> Data Sources -> New -> Generic Data Source, then enter:
   - Name: **DM_DS** or a similar value
   - JNDI Name: Enter the proper JNDI name.

     The JNDI name must be the JNDI name you specify in config.xml and in your Inline Service application parameter, CLMDS by default.
   - Database Type: **Oracle**

   Click Next.
7. For the Database Driver, select **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later**, then click Next.
8. In the Transaction Options window, deselect **Supports Global Transactions**, then click Next.
9. For the Connection Properties:
   - Specify the database settings that match your configuration, for example:



   Ensure that the Database User name and Password match the values that you set up in step 1 of Section 1.4.3.2, "Oracle RTD Decision Management Database Creation."
   - Click Next.
10. Leave all the settings already filled, except enter CHOICE for the Test Table Name, and click Test Configuration. Then click Next.

11. Select the Managed Server that you created in Section 1.4.2, "Production WebLogic Domain Creation" as the target, then click Finish.
12. Select **Activate Changes**.

1.4.3.4 Oracle RTD Decision Management Data Source Access by Oracle RTD Setup

Section 8.3.3 of *Oracle Real-Time Decisions Installation and Administration Guide* describes the general steps to add a data source to Oracle RTD. Follow the steps in that section, and, specifically for the Oracle RTD Decision Management data source, add the following in `web.xml` for `rtis.war`:

> **Note:** The code lines that follow show CLMDS as the value for `<res-ref-name>`. `<ref-ref-name>` refers to the JNDI data source name, which must be the JNDI name you specify in config.xml and in your Inline Service application parameter. Replace CLMDS, if necessary, with your particular JNDI name.

```
<resource-ref id="CLM_RTIS">
    <res-ref-name>CLMDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

and add the following in `web.xml` for `soap.war`:

```
<resource-ref id="CLM_Axis">
    <res-ref-name>CLMDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
     <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

1.4.3.5 Storing Credentials to Enable Web Service Calls

Oracle RTD Decision Management makes web service calls from the Decision Management server side to Oracle RTD. Credentials for these calls are stored in the WebLogic credential store. In summary form, the credential setup process is as follows:

- First, you create a user that will be used to authenticate the web service calls
- Then, you store the credentials of this user in the WebLogic credential store, so that the Decision Management application can retrieve them

To create the user and to store the user credentials in the WebLogic credential store, perform the following steps:

1. Open the Admin Server Console.

   On Windows, this is available via Start Menu: Programs -> Oracle WebLogic -> User Projects -> <RTDCLM_Prod_Domain> -> Admin Server Console.
2. Log in with the administrator username and password.
3. Select Security Realms and select the security realm you are using (myrealm by default).
4. Select the Users and Groups tab, and click New to create a new user.

5. Specify a username and password (generic names *<clmclient_user>* and *<clmclient_password>*).
6. Click OK.
7. Select the user that you just created and select the Groups tab.
8. Select the RTDDCUserGroup group, shuttle it to the right.
9. Click Save.
10. Open the Enterprise Manager on the Administration Server.
11. Log in with the administrator username and password.
12. In the Target Navigation Pane, select WebLogic Domain -> <RTDCLM_Prod_Domain>.
13. In the *<RTDCLM_Prod_domain>* window, from the WebLogic Domain dropdown menu, select Security -> Credentials.
14. In the Credentials area, click Create Map, and in the Create Map window, enter **oracle.rtd.clm** and click OK.
15. Click Create Key, and in the Create Key window:
    - Enter the Key **clm.client** for the map that you just created
    - Enter the same User Name and Password as you created previously in this process
    - Click OK

This completes the set up of the credentials for accessing Oracle RTD from Oracle RTD Decision Management.

1.4.3.6 Oracle RTD Decision Management Application Deployment

The final step is to deploy the Oracle RTD Decision Management application. You should have a clm.ear file that was generated from the development environment. The clm.ear application will be deployed to the production environment.

> **Note:** Enterprise Manager must be used to deploy the Oracle RTD Decision Management application, Weblogic Admin console cannot be used.

1. Start Node Manager.

   On Windows, Node Manager is either running as a service, or you can start it from the Start Menu: Programs -> Oracle WebLogic -> WebLogic 11gR1 -> Tools -> Node Manager.
2. Open the Enterprise Manager on the Administration Server.
3. Log in with the administrator username and password.
4. In the Target Navigation Pane, select WebLogic Domain -> <RTDCLM_Prod_Domain>.
5. In the <RTDCLM_Prod_Domain> window, from the WebLogic Domain dropdown menu, select Control -> Start Up.
6. In the <RTDCLM_Prod_Domain> window, from the WebLogic Domain dropdown menu, select Application Deployment -> Deploy....

7. In the Select Archive window, select the .ear file that was originally generated in the development environment.
8. In the Select Target window, select the managed server only.
9. In the Application Attributes window, do not alter any values.
10. In the Deployment Settings window, click Configure Application Security.
11. In the Configure Application Security window, for your first deployment, select Append for the Application Policy Migration and click Apply.

> **Note:** Use the Append options for first-time deployment only. For subsequent deployments:
>
> - For Application Policy Migration, select Overwrite.

12. Back in the Deployment Settings window, click Deploy.

    The Oracle RTD Decision Management application is now deployed and accessible at the URL: `http://<server>:<port>/dm`.

1.4.3.7 Inline Service Deployment

The application that you deployed requires a specific Inline Service to be running in the Oracle RTD instance on the same server as Oracle RTD Decision Management.

You should have received this Inline Service with the clm.ear file. Using Decision Studio, you must deploy that Inline Service to the Oracle RTD instance running on the same server as Oracle RTD Decision Management.

1.5 Installing Oracle RTD Decision Management for Production on WebSphere

This section contains the following topics:

- Section 1.5.1, "Preparing for Installation"
- Section 1.5.2, "Installing Oracle RTD Decision Management"

1.5.1 Preparing for Installation

To prepare for the installation of Oracle RTD Decision Management, perform the following steps *(if details are required, use the appropriate product documentation)*:

1. Install Oracle RTD for WebSphere following chapters 2 and 4 of the *Oracle Real-Time Decisions Installation and Administration Guide* (and any documentation specific to the version of Oracle RTD that you are installing).

> **Note:** Due to a limitation with Oracle Application Development Runtime on Windows, the path to the location where you install WebSphere cannot contain any spaces, therefore do not install in Program Files, which is the default location.

2. Install Application Development Runtime.
3. Install Enterprise Manager.

4. Set up the SOAP JVM custom property.
5. Modify the Oracle RTD Data Source.
6. Set up the DisableMultiThreadedServletConnectionMgmt custom property.

The rest of this section contains the following topics:

1.5.1.1 Setting Up SOAP JVM Custom Property

To set up the SOAP JVM custom property, perform the following steps:

1. Start WebSphere.
2. Login to the console with your user name and password.
3. From the left side of the console, expand Servers and choose Server types, then click WebSphere application servers.
4. Click the name of the application server where Oracle RTD is running (for example, server1).
5. Under Server Infrastructure, expand Java and Process Management, click Process definition, then select Java Virtual Machine.
6. Under Additional Properties, click Custom properties.
7. Click New, and under General properties, enter the following:
   - For Name, enter javax.xml.soap.SOAPConnectionFactory.
   - For Value, enter com.sun.xml.internal.messaging.saaj.client.p2p.HttpSOAPConnectionFactory.
8. Click OK, then click Save.
9. Restart the WebSphere application server.

1.5.1.2 Modifying the Oracle RTD Data Source

You must modify the Oracle RTD data source (SDDS) to enable it to work in this Oracle RTD Decision Management setup.

To modify the Oracle RTD data source, perform the following steps:

1. Login to the console with your user name and password.
2. From the left side of the Console window, expand Resources, then expand JDBC and choose Data sources.
3. Check RTD_DS and click Delete.
4. Click Save.
5. From the left side of the Console window, expand Resources, then expand JDBC and choose JDBC Providers.
6. Check RTDDataProvider and click Delete.
7. Click Save.

8. Still under JDBC providers, ensure that the scope is set correctly (to **Node=host_nameNode_*number***), then perform the following steps:
   - Click New.
   - For Database type, select Oracle.
   - For Provider type, select Oracle JDBC Driver.
   - For Implementation Type, select Connection Pool Data Source.
   - For Name, enter RTDDataProvider.
9. Click Next.
10. Click Next.
11. Click Finish, and then click Save.
12. Click on RTDDataProvider.
13. Change the class path to *RTD_HOME*/lib/jdbc/ojdbc14.jar.
14. Check Isolate this resource provider.
15. Click OK, and then click Save.
16. Click on RTDDataProvider.
17. Under the Additional Properties heading, click Data sources.
18. Click New, and enter the following values:
    - For Data source name, enter RTD_DS, or a similar value.
    - For JNDI name, enter SDDS.

    The JNDI name *must* be SDDS for Oracle RTD to access the database.
19. Click Next.
20. For URL, enter jdbc:oracle:thin:@*db_host*:*db_port*:*sid*.

    For example: jdbc:oracle:thin:@dbhost.company.com:1521:orcl.
21. For Data store helper class name, select Oracle 10g data store helper or Oracle 11g data store helper, depending on your Oracle RTD database.
22. Uncheck the option Use this data source in container managed persistence (CMP), and click Next.
23. For both Component-managed authentication alias and Container-managed authentication alias, select RTDDS_auth.

    Note: Do not change the values for Mapping-configuration alias.

24. Click Next.
25. Click Finish, and then click Save.
26. Restart WebSphere, then launch the Integrated Solutions Console again.
27. Expand Resources, then expand JDBC and choose Data Sources.
28. Select RTD_DS and click Test Connection.

    If the connection fails, ensure that your data source settings are correct, then test the data source again.

1.5.1.3 Setting Up DisableMultiThreadedServletConnectionMgmt Custom Property

To set up the DisableMultiThreadedServletConnectionMgmt custom property, perform the following steps:

1. Start WebSphere.
2. Login to the console with your user name and password.
3. From the left side of the console, expand Servers and choose Server types, then click WebSphere application servers.
4. Click the name of the application server where Oracle RTD is running (for example, server1).
5. Under Container Settings, expand Web Container Settings and click on Web container.
6. Under Additional Properties, click Custom properties.
7. Click New, and under General properties, enter the following:
   - For Name, enter DisableMultiThreadedServletConnectionMgmt
   - For Value, enter true
8. Click OK, then click Save.
9. Restart the WebSphere application server.

1.5.2 Installing Oracle RTD Decision Management

After you have created the WebSphere Server and configured Oracle RTD to run on this server, you are ready to install Oracle RTD Decision Management.

This section contains the following topics:

1.5.2.1 Files Required from Development Environment

The Oracle RTD Decision Management configuration performed in the development environment is the source of the following files, required for the production environment:

- clm.ear
- load core.sql
- load ils.sql
- The Inline Service that is part of this application (see Section 1.2.2.5, "Inline Service Deployment")

Note: The .sql files are generated when you run ant generate (and before you run ant ear).

1.5.2.2 Oracle RTD Decision Management Database Creation

Note: For running the SQL commands and SQL scripts specified in this section, use SQL*Plus or an equivalent tool that can run on your database.

To create an Oracle RTD Decision Management database, perform the following steps:

1. Create a database user for the Oracle RTD Decision Management schema.

```
CREATE USER <username> IDENTIFIED BY <password> DEFAULT TABLESPACE users
TEMPORARY TABLESPACE temp QUOTA UNLIMITED ON users;

GRANT CREATE VIEW, CONNECT, RESOURCE TO <username>;
```

Note: If you are going to use non 7-bit ASCII characters in choice group id or choice attribute ids, you must set the environment variable NLS_LANG to .UTF8 before running the sql scripts in the steps that follow.

2. Create the schema, by performing the following steps:
   a. Go to the subfolder clm\Database\sql of the root folder where you unzipped RTD_Designer.zip was unzipped, then run the file load core.sql.

Note: The file load core.sql contains generic commands, required for all Oracle RTD Decision Management applications. The commands in load ils.sql, as released, are for the RTD for Marketing Optimization application.

   b. Go to the subfolder clm\Database\sql\ils of the folder where you unzipped RTD_Designer.zip was unzipped, then run the file load ils.sql.

      Make sure you have committed your changes.

1.5.2.3 Oracle RTD Decision Management Data Source Setup

Use the Integrated Solutions Console to create a JDBC provider for the Oracle RTD Decision Manager Database.

Note: Before you begin, ensure that WebSphere is started.

To create a JDBC provider for the Oracle RTD Decision Manager Database, perform the following steps:

1. Login to the console with your user name and password.
2. From the left side of the console window, expand Resources, then expand JDBC and choose JDBC Providers.

3. Ensure that the scope is set correctly (to **Node=*host_name*Node_*number***), then click New, and select or enter the following:
   - For Database type, select Oracle.
   - For Provider type, select Oracle JDBC Driver.
   - For Implementation Type, select Connection Pool Data Source.
   - For Name, enter DMDataProvider.
4. Click Next.
5. Click Next.
6. Click Finish, and then click Save.
7. Click on DMDataProvider.
8. In the General properties of the page, replace the value in the Class path field with the following:
   - ${COMMON_COMPONENTS_HOME}/modules/oracle.jdbc_11.1.1/ojdbc6dms.jar
   - ${COMMON_COMPONENTS_HOME}/modules/oracle.dms_11.1.1/dms.jar
   - ${COMMON_COMPONENTS_HOME}/modules/oracle.odl_11.1.1/ojdl.jar

   Note: Press Enter to separate the path locations so they appear on separate lines.
9. Click OK, to return to the JDBC Providers page.
10. Click Save to save your changes to the master configuration.
11. From the left side of the Console window, choose Security, then Global Security.
12. Under the Authentication section in the right column, expand the Java Authentication and Authorization Service option.
13. Choose the J2C Authentication data option.
14. To create a new authentication data entry, click New, and enter the following values:
    - For Alias, enter DMDS_auth.
    - For User ID, enter the name of the database run-time user.
    - For Password, enter the corresponding password for the database user.
15. Click OK, then click Save
16. From the left side of the Console window, choose JDBC, then JDBC providers.
17. On the JDBC providers page, click DMDataProvider. Then, under the Additional Properties heading, click Data sources.
18. Click New, and enter the following values:
    - For Data source name, enter DM_DS, or a similar value.
    - For JNDI name, enter the proper JNDI name. The JNDI name *must* be the JNDI name you specify in config.xml and in your Inline Service application parameter, CLMDS by default.

19. Click Next.
20. For URL, enter jdbc:oracle:thin:@*db_host*:*db_port*:*sid*.

    For example: `jdbc:oracle:thin:@dbhost.company.com:1521:orcl`.
21. For Data store helper class name, select Oracle 10g data store helper or Oracle 11g data store helper, depending on your Oracle RTD Decision Management database.
22. Uncheck the option Use this data source in container managed persistence (CMP), and click Next.
23. For both Component-managed authentication alias and Container-managed authentication alias, select DMDS_auth.

    Note: Do not change the value for Mapping-configuration alias.

24. Click Next.
25. Click Finish, and then Save.
26. Click your Decision Manager data source, for example, DM_DS.
27. Under Additional Properties, click WebSphere Application Server data source properties.
28. Select Non-transactional data source.
29. Click OK, then click Save.
30. Restart WebSphere, then launch the Integrated Solutions Console again.
31. Expand Resources, then expand JDBC and choose Data Sources.
32. Select DM_DS and click Test Connection.

    If the connection fails, ensure that your data source settings are correct, then test the data source again.

1.5.2.4 Oracle RTD Decision Management Data Source Access by Oracle RTD Setup

Section 8.2.2 of Oracle Real-Time Decisions Installation and Administration Guide describes the general steps to add a data source to Oracle RTD. Follow the steps in that section, and, specifically for the Oracle RTD Decision Management data source, add the following in `web.xml` for `rtis.war`:

Note: The code lines that follow show CLMDS as the value for `<res-ref-name>`. `<ref-ref-name>` refers to the JNDI data source name, which must be the JNDI name you specify in config.xml and in your Inline Service application parameter. Replace CLMDS, if necessary, with your particular JNDI name.

```
<resource-ref id="CLM_RTIS">
    <res-ref-name>CLMDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
    <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

and add the following in `web.xml` for `soap.war`:

```
<resource-ref id="CLM_Axis">
    <res-ref-name>CLMDS</res-ref-name>
    <res-type>javax.sql.DataSource</res-type>
    <res-auth>Container</res-auth>
     <res-sharing-scope>Unshareable</res-sharing-scope>
</resource-ref>
```

After you add resource references to the `web.xml` files, you must configure the resource references in WebSphere.

Follow the steps in Section 8.2.3 of Oracle Real-Time Decisions Installation and Administration Guide, and for the Target Resource JNDI Name, use the JNDI Name that you specified for your data source.

1.5.2.5 Storing Credentials to Enable Web Service Calls

Oracle RTD Decision Management makes web service calls from the Decision Management server side to Oracle RTD. Credentials for these calls are stored in the WebSphere credential store. In summary form, the credential setup process is as follows:

- First, you create a user that will be used to authenticate the web service calls.
- Then, you store the credentials of this user in the WebSphere credential store, so that the Decision Management application can retrieve them.

To create the user and to store the user credentials in the WebSphere credential store, perform the following steps:

1. Access the Integrated Solutions Console at the URL `http://<websphere_host>:<port>/ibm/console`. On Windows, you can also access the Integrated Solutions Console through Start > Programs.
2. At the login prompt, enter the administrator user name and password.
3. In the tree on the left, expand Users and Groups and select Manage Users.
4. Click Create.
5. In the Create a User form, specify a User ID and Password (generic names *<clmclient_user>* and *<clmclient_password>*).
6. Confirm the password and optionally enter other information about user.
7. Click Group Membership.
8. In the Group Membership form, add RTDDCUserGroup to the Current groups, then click Close.
9. Back in the Create a User form, click Create.

To store user credentials for web service calls, perform the following steps:

1. Open the Enterprise Manager.
2. Log in with the administrator username and password.
3. Select WebSphere Cell, then Security, then Credentials.
4. In the Credentials area, click Create Map, and in the Create Map window, enter oracle.rtd.clm and click OK.
5. Click Create Key, and in the Create Key window:

   Enter the Key clm.client for the map that you just created

Enter the same User Name (<clmclient_user>) and Password (<clmclient_password>) as the User Id and Password that you created previously in this process.

Click OK.

1.5.2.6 Oracle RTD Decision Management Application Deployment

The final step is to deploy the Oracle RTD Decision Management application. You should have a clm.ear file that was generated from the development environment. The clm.ear application will be deployed to the production environment.

To deploy the Oracle RTD Decision Management application, perform the following steps:

1. Login to the console with your user name and password.
2. From the left side of the Console window, expand Applications, and then choose Application types, then WebSphere enterprise applications.
3. Click Install.
4. In the Path to the new application section, enter or browse to select the clm.ear file.
5. Click Next.
6. Click Next again.
7. Accept the default settings of the Select installation options section, and click Next.
8. Accept the default settings of the Map modules to servers section, and click Next.
9. Click Finish, and then Save.
10. Click the Enterprise Application Name clm *(unless you modified this name in config.xml)*.
11. Under Modules, click Manage Modules.
12. Click Oracle Real-Time Decisions Manager.
13. Under Additional Properties, click Session Management.
14. Select and Open Enable Cookies.
15. Enter cookie path: /dm *(if you chose a different context root then specify it here)*.
16. Click OK.
17. Select Override session management.
18. Click OK twice.
19. Click Save to save it directly to the master configuration.
20. Open Applications, then Application Types, then WebSphere enterprise application and start clm.

    The Oracle RTD Decision Management application is now deployed and accessible at the URL: http://<server>:<port>/dm.

1.5.2.7 Inline Service Deployment

The application that you deployed requires a specific Inline Service to be running in the Oracle RTD instance on the same server as Oracle RTD Decision Management.

You should have received this Inline Service with the clm.ear file. Using Decision Studio, you must deploy that Inline Service to the Oracle RTD instance running on the same server as Oracle RTD Decision Management.

1.5.2.8 Setting HttpOnly Flag

For additional security, you can optionally restrict the Decision Manager cookies to HttpOnly, to prevent malicious JavaScript code accessing them.

To set the HttpOnly flag, perform the following steps:

1. Start WebSphere.
2. Login to the console with your user name and password.
3. From the left side of the console, expand Servers and choose Server types, then click WebSphere application servers.
4. Click the name of the application server where Oracle RTD is running (for example, server1).
5. Under Container Settings, expand Web Container Settings and click on Web container.
6. Under Additional Properties, click Custom properties.
7. Click New, and under General properties, enter the following:
   - For Name, enter com.ibm.ws.webcontainer.HTTPOnlyCookies
   - For Value, enter *
8. Click OK, then click Save.
9. Restart the WebSphere application server.

1.6 Security Configuration

There are three ways to setup authentication for Oracle RTD Decision Management:

- In the development environment on WebLogic, you can manage the users within JDeveloper.
- In the production environment on WebLogic, you can map your enterprise users and groups to Decision Management application roles using Enterprise Manager.
- In both development environment and production environments, on both WebLogic and WebSphere, you can map your enterprise users and groups to Decision Management application roles using intermediary enterprise roles.

This section also describes how to setup SSL to secure connections between the end user browser and the application server. Setting up SSL is not required if you do not wish to use this encryption.

This section also describes how to setup SSO to enable integration of Decision Center reports within Decision Manager. Setting up SSO is not required if you do not wish to see these reports within Decision Manager and will use Decision Center for viewing the reports.

This section contains the following topics:

1.6.1 Managing Users in JDeveloper in a Development Environment under WebLogic

The simplest way to configure access to a Decision Management application in development mode is to specify users in Oracle RTD Decision Management metadata and map them to Decision Management application roles in Oracle RTD Decision Management metadata. When deploying from JDeveloper, this will automatically create the users in WebLogic and map them to the Decision Management application roles.

This is done with these steps:

- Retrieve the encrypted password for your user
- Add the user in Oracle RTD Decision Management metadata and map the user to Decision Management application roles
- Run application generation
- Deploy the Decision Management application
- Add the user to Oracle RTD groups

This section contains the following topics:

1.6.1.1 Retrieving the Encrypted Password for Your User

Passwords are stored encrypted in `jazn-data.xml`. You will need to access the encrypted password and add the encrypted password explicitly to Oracle RTD Decision Management metadata in a subsequent setup step.

You can use JDeveloper to get the encrypted password string, as follows:

1. Open JDeveloper from Start Menu, Programs -> Oracle Fusion Middleware *<11g_version>* -> JDeveloper Studio <11g_version>.
2. Select Default Role.

3. Click Open Application... and select **clm.jws** in the clm subfolder of where you unzipped `RTD_Designer.zip`.
4. In Application Navigator, open Application Resources.
5. Expand Descriptors -> META-INF, and double-click **jazn-data.xml**.
6. In the Users tab, create a user by clicking on the + icon next to Users.
7. Enter a user Name and a Password.
8. Click the Source tab (from the tab list at the bottom of the window).

   In the source XML, you can now locate the credentials for the user that you just created. The credentials value is the encrypted password for the user.

   For example, after entering user name clmuser and a password for clmuser, this could appear in the Source *(with the encrypted password highlighted in bold in the example below)* as:

```
<user>
    <name>clmuser</name>
    <credentials>{903}loyIIwIDNKqdw4D9XrMhQDZ4yEFLXim+</credentials>
</user>
```

9. Copy the encrypted password into your standard machine buffer (Ctrl-C).

   Retain it there for use in the subsequent setup step, Section 1.6.1.2, "Adding Users in Metadata and Mapping Them to Application Roles.".

1.6.1.2 Adding Users in Metadata and Mapping Them to Application Roles

The description in this section assumes that you are editing the Decision Manager for Marketing Optimization implementation. If you use a different application, see Chapter 2, "Configuring Oracle RTD Decision Management" for the differences.

To add users in Oracle RTD Decision Management metadata and to map them to Decision Management application roles, perform the following steps:

1. Go to the `clm/Build/metadata/ref` folder.
2. Edit `security.xml`, or create a different file, such as `users.xml`.

   Add user information (including the encrypted password that you saved in the previous setup step) using the following as an example (this example assigns the **CLMAdministrator** application role to the user **clmuser**):

```
<users>
    <user>
        <name>clmuser</name>
        <display-name>clmuser</display-name>
        <description>this is a clm user</description>
        <credentials>{903}loyIIwIDNKqdw4D9XrMhQDZ4yEFLXim+</credentials>
        <application-roles>CLMAdministrator</application-roles>
    </user>
</users>
```

1.6.1.3 Running Application Generation

To run application generation, perform the following steps

1. Open a command prompt in the clm/Build directory.
2. Run the command **`ant generate -Dchanges=overwrite`**.

Note: You must overwrite changes because you made changes to `jazn-data.xml` in the previous setup step when you manually entered a password in JDeveloper, and here the ant generation overwrites this file.

In JDeveloper, verify that the user now shows up in `jazn-data.xml` and is mapped to the CLMAdministrator application role.

1.6.1.4 Deploying the Oracle RTD Decision Management Application

In JDeveloper, deploy the Decision Management application to AdminServer by choosing menu item: Application -> Deploy -> clm_application1.

1.6.1.5 Adding Users to Oracle RTD Groups

After the user has been created in WebLogic, you must add the user to the RTDDCUserGroup and the RTDChoiceEditorGroup groups, by performing the following steps:

1. Open the Admin Server Console.

   On Windows, this is available via the Start Menu: Programs -> Oracle Fusion Middleware <11g_version> -> User Projects -> *<RTDCLM_Dev_Domain>* -> Admin Server Console.

2. Log in with the administrator username and password.
3. Select Security Realms and select the security realm you are using (myrealm by default).
4. Select the Users and Groups tab.
5. Click the user that you created in Section 1.6.1.1, "Retrieving the Encrypted Password for Your User."
6. Select the Groups tab.
7. Select RTDDCUserGroup and RTDChoiceEditorGroup, and shuttle them to the right.
8. Click Save.

1.6.2 Mapping Users or Groups to Application Roles for Production under WebLogic

In a WebLogic production environment, you typically manage your enterprise users and groups outside of Oracle RTD Decision Management, and only map these users and groups to Decision Management application roles.

In this section, we will use "clmuser" and "clmgroup" as examples of your enterprise users and groups.

This section contains the following topics:

- Section 1.6.2.1, "Mapping Users or Groups to Decision Management Application Roles"
- Section 1.6.2.2, "Adding the User or Group to Oracle RTD Groups"

1.6.2.1 Mapping Users or Groups to Decision Management Application Roles

To map users or groups to Decision Management application roles, perform the following steps:

1. Open the Enterprise Manager on the Administration Server.
2. Log in with the administrator username and password.
3. In the Target Navigation Pane, select the clm deployment: Application Deployments -> Internal Applications -> clm.
4. In the clm window, from the Application Deployment dropdown menu, select Security -> Application Roles and click the "Search application roles" button.
5. Click CLMAdministrator.
6. Click Edit the selected application role.
7. Click Add roles.
8. Change Application role to User in the Type dropdown list.
9. Search for your user and click OK.
10. To add a group, click Add roles, change Application role to Group in the Type dropdown list.
11. Search for your group and click OK.
12. Click OK.

1.6.2.2 Adding the User or Group to Oracle RTD Groups

To use Oracle RTD Decision Management, your user or groups must be in the RTDDCUserGroup and RTDChoiceEditorGroup groups.

To add users or groups to the Oracle RTD groups, perform the following steps:

1. Open the Admin Server Console.

   On Windows, this is available via Start Menu: Programs -> Oracle WebLogic -> User Projects -> <RTDCLM_Prod_Domain> -> Admin Server Console.
2. Log in with the administrator username and password.
3. Select Security Realms and select the security realm you are using (myrealm by default).
4. Select the Users and Groups tab.
5. Click **clmuser**.
6. Select the Groups tab.
7. Select RTDDCUserGroup and RTDChoiceEditorGroup, and shuttle them to the right.
8. Go back to the Users and Groups tab.
9. Select the Groups tab.
10. Click **clmgroup**, and click the Membership tab.
11. Select RTDDCUserGroup and RTDChoiceEditorGroup, and shuttle them to the right.
12. Click Save.

1.6.3 Mapping Users and Groups to Intermediary Roles in Development

If you want to use your enterprise users and groups in development mode, you specify enterprise roles in Decision Management metadata and map them to Decision

Management application roles. When deploying from JDeveloper, this will automatically create groups in WebLogic for these enterprise roles and map them to the Decision Management application roles.

This is done with these steps:

- Add the enterprise role in Decision Management metadata and map it to Decision Management application roles
- Run application generation
- Deploy the Oracle RTD Decision Management application
- Specify this group as a member of Oracle RTD groups
- Map your enterprise users or groups to this Decision Management group

This section contains the following topics:

1.6.3.1 Mapping the Enterprise Role to Decision Management Application Roles

The description in this section assumes that you are editing the RTD for Marketing Optimization implementation. If you use a different application, see Chapter 2, "Configuring Oracle RTD Decision Management" for the differences.

To add the enterprise role in Oracle RTD Decision Management metadata and to map it to Decision Management application roles, perform the following steps:

1. Go to the `clm/Build/metadata/ref` folder.
2. Edit `security.xml`, or create a different file, such as `groups.xml`.

   Add the following (here we create a **clmgroup** enterprise role and map it to the **CLMAdministrator** application role):

```
<enterprise-roles>
    <enterprise-role>
        <name>clmgroup</name>
        <application-roles>CLMAdministrator</application-roles>
    </enterprise-role>
</enterprise-roles>
```

1.6.3.2 Running Decision Management Application Generation

To run application generation, perform the following steps

1. Open a command prompt in the clm/Build directory.
2. Run the command **`ant generate`**.

In JDeveloper, verify that the group now shows up in `jazn-data.xml` and is mapped to the CLMAdministrator application role.

1.6.3.3 Deploying the Decision Management Application

In JDeveloper, deploy the Decision Management application to AdminServer by choosing the menu item: Application -> Deploy -> clm_application1.

1.6.3.4 Specifying the WebLogic Group as a Member of Oracle RTD Groups

After the enterprise role has been created as a group in WebLogic, you must make it a member of the RTDDCUserGroup and the RTDChoiceEditorGroup groups, by performing the following steps:

1. Open the Admin Server Console.

   On Windows, this is available via the Start Menu: Programs -> Oracle Fusion Middleware <11g_version> -> User Projects -> *<RTDCLM_Dev_Domain>* -> Admin Server Console.

2. Log in with the administrator username and password.
3. Select Security Realms and select the security realm you are using (myrealm by default).
4. Select the Users and Groups tab.
5. Select the Groups tab.
6. Click **clmgroup**.
7. Click the Membership tab.
8. Select RTDDCUserGroup and RTDChoiceEditorGroup, and shuttle them to the right.
9. Click Save.

1.6.3.5 Mapping Enterprise Users or Groups to the Decision Management Group

You can now grant access to Oracle RTD Decision Management to your users by adding them the **clmgroup** group.

You can also grant access to Oracle RTD Decision Management to your groups by making them a member of the **clmgroup** group.

1.6.4 Mapping Users and Groups to Intermediary Roles in Production

One way to use your enterprise users and groups in production mode on WebLogic or WebSphere is to specify enterprise roles in Decision Management metadata and map them to Decision Management application roles. Create groups in WebLogic or WebSphere for these enterprise roles and, after deploying the Decision Manager enterprise application using the WebLogic or WebSphere console, these groups will be mapped to the corresponding Decision Management application roles.

This is done with these steps:

- Add the enterprise role in Decision Management metadata and map it to Decision Management application roles
- Run application generation
- Create the group and specify it as a member of Oracle RTD groups
- Deploy the Oracle RTD Decision Management application
- Map your enterprise users or groups to this Decision Management group

This section contains the following topics:

- Section 1.6.4.1, "Mapping the Enterprise Role to Decision Management Application Roles"
- Section 1.6.4.2, "Running Decision Management Application Generation"
- Section 1.6.4.3, "Creating the Group and Specifying it as a Member of Oracle RTD Groups"
- Section 1.6.4.4, "Deploying the Decision Management Application"
- Section 1.6.4.5, "Mapping Enterprise Users or Groups to the Decision Management Group"

1.6.4.1 Mapping the Enterprise Role to Decision Management Application Roles

The description in this section assumes that you are editing the RTD for Marketing Optimization implementation. If you use a different application, see Chapter 2, "Configuring Oracle RTD Decision Management" for the differences.

To add the enterprise role in Oracle RTD Decision Management metadata and to map it to Decision Management application roles, perform the following steps:

1. Go to the `clm/Build/metadata/ref` folder.
2. Edit `security.xml`, or create a different file, such as `groups.xml`.

   Add the following (here we create a **clmgroup** enterprise role and map it to the **CLMAdministrator** application role):

```
<enterprise-roles>
    <enterprise-role>
        <name>clmgroup</name>
        <application-roles>CLMAdministrator</application-roles>
    </enterprise-role>
</enterprise-roles>
```

1.6.4.2 Running Decision Management Application Generation

To run application generation, perform the following steps

1. Open a command prompt in the clm/Build directory.
2. Run the command **`ant generate`**.

In JDeveloper, verify that the group now shows up in `jazn-data.xml` and is mapped to the CLMAdministrator application role.

1.6.4.3 Creating the Group and Specifying it as a Member of Oracle RTD Groups

Using the WebLogic or WebSphere console, create a group with the same name as the enterprise role, for example, clmgroup, and make it a member of the RTDDCUserGroup and the RTDChoiceEditorGroup groups.

1.6.4.4 Deploying the Decision Management Application

For details of deployment on Weblogic, see Section 1.4.3.6, "Oracle RTD Decision Management Application Deployment."

For details of deployment on WebSphere, see Section 1.5.2.6, "Oracle RTD Decision Management Application Deployment."

1.6.4.5 Mapping Enterprise Users or Groups to the Decision Management Group

You can now grant access to Oracle RTD Decision Management to your users by adding them the **clmgroup** group.

You can also grant access to Oracle RTD Decision Management to your groups by making them a member of the **clmgroup** group.

1.6.5 Using SSL with Oracle RTD Decision Management Applications on WebLogic

This section, which is for WebLogic environments, contains the following topics:

- Section 1.6.5.1, "SSL for Development Environment"
- Section 1.6.5.2, "SSL for Production Environment"

1.6.5.1 SSL for Development Environment

To set up SSL for the development environment, perform the following steps:

1. Configure keystore and trust store as described in Chapter 2.6, "Using SSL with Oracle Real-Time Decisions" in *Oracle Real-Time Decisions Installation and Administration Guide*.
2. Perform the instructions in Section 5.13, "Configuring SSL for Real-Time Decision Server" of *Oracle Real-Time Decisions Installation and Administration Guide*.

   The remainder of this sequence assumes that you have successfully installed the Oracle RTD certificates and can access the non-SSL ports, but does not assume that you have disabled the SSL ports.

   Please note that the Oracle RTD certificates are for testing only and should not be used in production.
3. Open the Admin Server Console.

   On Windows, this is available via the Start Menu: Programs -> Oracle Fusion Middleware <11g_version> -> User Projects -><*RTDCLM_Dev_domain*> -> Admin Server Console.
4. Log in with the administrator username and password.
5. Navigate the path Environments > Servers > AdminServer.
6. Select AdminServer, then Configuration tab -> SSL tab > Advanced.
7. Select "Use Server Certs". Save your changes to the SSL tab.
8. Add the following to <*domain*>\bin\setDomainEnv.cmd:

   -Djavax.net.ssl.trustStore=<*RTD_HOME*>\RTD\etc\ssl\sdtrust.store

   or (Unix)

   -Djavax.net.ssl.trustStore=<*RTD_HOME*>/RTD/etc/ssl/sdtrust.store
9. Change clm\Build\metadata\<*metadata_module*>\config\config.xml workbenchClient to https://<*server*>:<*SSL port*>.
10. Run ant generate.
11. Stop and delete the existing Oracle RTD Decision Management application deployment.
12. Redeploy the application from JDeveloper using the connection you set up in Section 1.2.2.7, "Oracle RTD Decision Management Application Deployment."

13. Verify that both the Admin Server Console and the Oracle RTD Decision Management application are accessible using the SSL port. After you have determined this, you should consider disabling the non-SSL Admin Server port.

    Ignore the certificate errors from the browser. Note again that the Oracle RTD test certificate is for testing only.

14. Stop and restart all services.

15. Access Oracle RTD Decision Management through the appropriate secure port, for example, `https://myserver:7002/dm`.

1.6.5.2 SSL for Production Environment

To set up SSL for the production environment, perform the following steps:

1. Configure keystore and trust store as described in Chapter 2.6, "Using SSL with Oracle Real-Time Decisions" in *Oracle Real-Time Decisions Installation and Administration Guide*.

2. Perform the instructions in Section 5.13, "Configuring SSL for Real-Time Decision Server" of *Oracle Real-Time Decisions Installation and Administration Guide*.

   The remainder of this sequence assumes that you have successfully installed the Oracle RTD certificates and can access the non-SSL ports, but does not assume that you have disabled the SSL ports.

   Please note that the Oracle RTD certificates are for testing only and should not be used in production.

3. Open the Admin Server Console.

   On Windows, this is available via the Start Menu: Programs -> Oracle Fusion Middleware <11g_version> -> User Projects -><*RTDCLM_Prod_domain*> -> Admin Server Console.

4. Log in with the administrator username and password.

5. Navigate the path Environments > Servers > <*Managed_Server*>.

6. Select the Managed Server, then Configuration tab -> SSL tab > Advanced.

7. Select "Use Server Certs". Save your changes to the SSL tab.

8. Add the following to <*domain*>\bin\setDomainEnv.cmd:

   -Djavax.net.ssl.trustStore=<*RTD_HOME*>\RTD\etc\ssl\sdtrust.store

   or (Unix)

   -Djavax.net.ssl.trustStore=<*RTD_HOME*>/RTD/etc/ssl/sdtrust.store

   Alternatively, add this to the managed server "Server Start" -> "Arguments" box.

9. Change clm\Build\metadata\<*metadata_module*>\config\config.xml workbenchClient to https://<*server*>:<*SSL port*>.

10. Run ant generate.

11. Stop and delete the existing Oracle RTD Decision Management application deployment.

12. Redeploy the application to non-SSL port. Do this by using Enterprise Manager deployed on Admin Server and listening non-SSL port.

13. Verify that the Oracle RTD Decision Management application is accessible using the SSL port.

Ignore the certificate errors from the browser. Note that the Oracle RTD test certificate is for testing only.

14. Stop and restart all services.
15. Access Oracle RTD Decision Management through the appropriate secure port, for example, `https://myserverx:7002/dm`.

1.6.6 Using SSL with Oracle RTD Decision Management Applications on WebSphere

SSL is enabled by default when installing WebSphere, so you will be able to use Decision Manager using SSL out of the box.

Note: You can see the default port for SSL in the WebSphere Administration Console.

For example: Servers > Server Types > WebSphere Application servers > server1 > Communications > Ports > WC_defaulthost_ secure 9443.

For a proper configuration, you must acquire a server certificate for your domain and configure WebSphere to use the certificate. Refer to the WebSphere documentation for more details.

1.6.7 Using SSO with Oracle RTD Decision Management Applications on WebLogic

This section contains the following topics:

- Section 1.6.7.1, "Configuration Tasks for Oracle RTD Decision Management"
- Section 1.6.7.2, "Understanding SSO Authentication and Oracle RTD Decision Management"
- Section 1.6.7.3, "Prerequisites"
- Section 1.6.7.4, "Creating Groups and Users in LDAP Server"
- Section 1.6.7.5, "Installing and Configuring Oracle Access Manager (OAM) Client Software"
- Section 1.6.7.6, "Configuring Domain for Oracle RTD and Decision Manager"
- Section 1.6.7.7, "Troubleshooting"

1.6.7.1 Configuration Tasks for Oracle RTD Decision Management

This section provides a summary reference of the tasks required to set up single sign-on (SSO) solution with Oracle RTD Decision Management on WebLogic:

1. Installing and configuring Oracle Identity Management 11g and Oracle Access Manager (OAM) 11g.

   For details of how to install and configure these components, see *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

2. Creating Groups and Users in LDAP Server.
3. Installing and Configuring Oracle Access Manager (OAM) Client Software.
4. Configuring Domain for Oracle RTD and Decision Manager.

This section also includes a Troubleshooting section that describes issues that may be encountered during installation and configuration.

1.6.7.2 Understanding SSO Authentication and Oracle RTD Decision Management

Integrating a single sign-on (SSO) solution enables a user to log on (sign-on) and be authenticated once. Thereafter, the authenticated user is given access to system components or resources according to the permissions and privileges granted to that user. Oracle RTD Decision Management can be configured to trust incoming HTTP requests authenticated by a SSO solution that is configured for use with Oracle Fusion Middleware and Oracle WebLogic Server. For more information about configuring SSO for Oracle Fusion Middleware, see "Configuring Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*.

When Oracle RTD Decision Management is configured to use SSO authentication, it accepts authenticated users from whatever SSO solution OAM Webgate is configured to use. If SSO is not enabled, then Oracle RTD Decision Management challenges each user for authentication credentials. When Oracle RTD Decision Management is configured to use SSO, a user is first redirected to the SSO solution's login page for authentication. After the user is authenticated the SSO solution forwards the user name to Decision Manager where this name is extracted.

Configuring Oracle RTD Decision Management to work with SSO authentication requires minimally that Oracle Fusion Middleware and Oracle WebLogic Server are configured to accept SSO authentication.

How an Identity Asserter Works

This section describes how Oracle Access Manager authentication provider works with Oracle WebLogic Server using Identity Asserter for single sign-on, providing the following features:

- **Identity Asserter for Single Sign-on**

  This feature uses the Oracle Access Manager authentication services and validates already-authenticated Oracle Access Manager users through a suitable token and creates a WebLogic-authenticated session. It also provides single sign-on between WebGate and portals. WebGate is a plug-in that intercepts Web resource (HTTP) requests and forwards them to the Access Server for authentication and authorization.

- **Authenticator**

  This feature uses Oracle Access Manager authentication services to authenticate users who access an application deployed in Oracle WebLogic Server. Users are authenticated based on their credentials, for example a user name and password.

After the authentication provider for Oracle Access Manager is configured as the Identity Asserter for single sign-on, the Web resources are protected. Perimeter authentication is performed by WebGate on the Web tier and by the appropriate token to assert the identity of users who attempt access to the protected WebLogic resources.

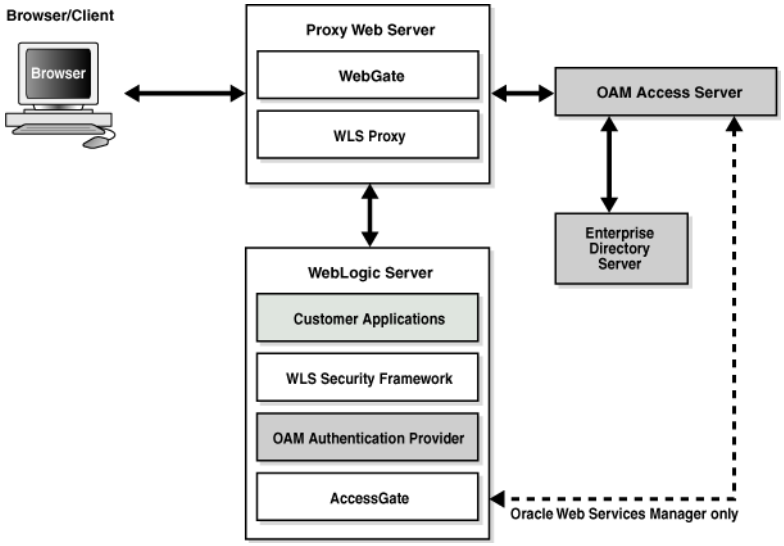
All access requests are routed to a reverse proxy Web server. These requests are in turn intercepted by WebGate. The user is challenged for credentials based on the authentication scheme configured within Oracle Access Manager (form-based login recommended).

After successful authentication, WebGate generates a token and the Web server forwards the request to Oracle WebLogic Server, which in turn invokes Oracle Access Manager Identity Asserter for single sign-on validation. The WebLogic Security Service invokes Oracle Access Manager Identity Asserter for single sign-on, which

next gets the token from the incoming request and populates the subject with the WLSUserImpl principal. The Identity Asserter for single sign-on adds the WLSGroupImpl principal corresponding to the groups the user is a member of. Oracle Access Manager then validates the cookie.

Figure 1–1 depicts the distribution of components and the flow of information when the Oracle Access Manager Authentication Provider is configured as an Identity Asserter for SSO with Oracle Fusion Middleware.

Figure 1–1 Oracle Access Manager Single Sign-On Solution for Web Resources Only



How Decision Manager Operates with SSO Authentication

After a user logs into Decision Manager, the SSO cookies are used to display parts of the Oracle RTD web application as integrated parts of the user interface.

SSO Implementation Considerations

When implementing a SSO solution with Oracle RTD Decision Management you should consider the following:

- As WebLogic server trusts Webgate to provide authentication information, it is essential to secure the machines that communicate directly with Decision Manager. Possible solutions are network-level router settings, or WebLogic's connection filtering, or mutual SSL authentication.

1.6.7.3 Prerequisites

This section provides a checklist of the components that are required for the installations and configurations in the sections that follow.

Oracle Access Manager 11g must be properly installed and configured.

Oracle Identity Management 11g must be installed.

Oracle Web Tier Utilities 11g and Oracle Webgate 11g must be installed.

For details of how to install and configure these products, see *Oracle Fusion Middleware Installation Guide for Oracle Identity Management* and *Oracle Fusion Middleware Installation Guide for Oracle Web Tier*.

Note: 1. The information described in this chapter assumes that Oracle Internet Directory is running and the proper naming context is configured. If your deployment is going to use a different LDAP solution, align your settings accordingly.

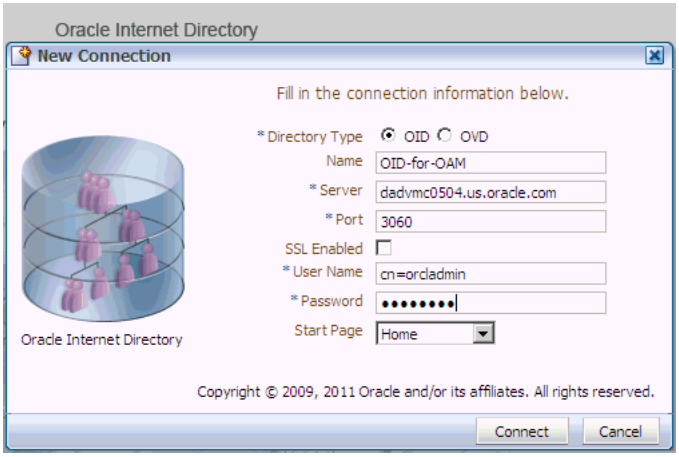
2. Oracle recommends that you register the Web Tier management component in the same WebLogic domain (referred to as DM_domain) as where Oracle RTD and the Decision Manager web application are to be deployed. This can be achieved by first choosing the installation type for Web Tier to be "Do Not Configure," and then configuring Oracle HTTP Server (OHS).

1.6.7.4 Creating Groups and Users in LDAP Server

Oracle Directory Services Manager (ODSM) provides a graphical administrative interface for Oracle Internet Directory. You can invoke ODSM directly from its own URL or from the Oracle Enterprise Manager Fusion Middleware Control pages for Oracle Internet Directory.

Use ODSM to create groups and users in LDAP Server by performing the following steps:

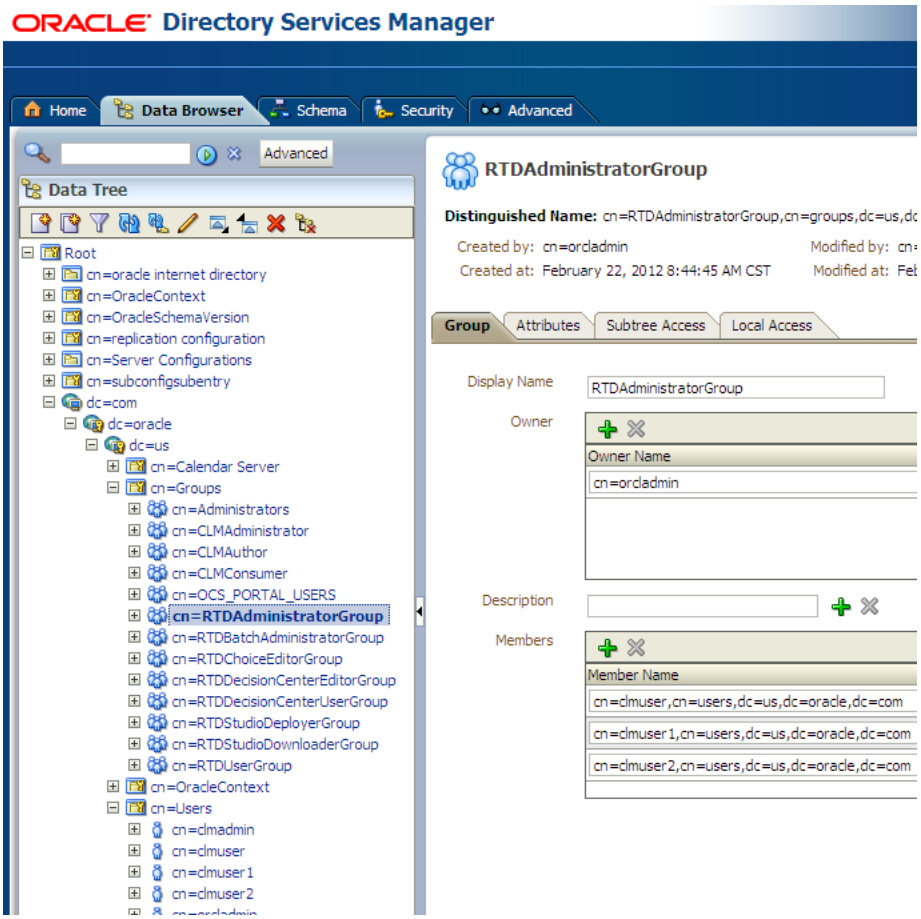
1. Open ODSM and create a new connection, as in the following example:



2. Open the Data Browser tab.

3. Create the required users and the following groups:

```
RTDUserGroup
RTDAdminGroup
RTDBatchAdminGroup
RTDChoiceEditorGroup
RTDDCEditorGroup
RTDDCUserGroup
RTDStudioDeployerGroup
RTDStudioDownloaderGroup
CLMAdministrator
CLMAuthor
CLMConsumer
```



4. Add the required users as members to the appropriate groups.

> **Note:** The rtdadmin user belonging to the RTDAdminGroup must be created.

1.6.7.5 Installing and Configuring Oracle Access Manager (OAM) Client Software

> **Note:** Production deployment topology considerations generally result in different machines to host OAM server, Web Tier, and SSO-enabled applications. To avoid confusion, the topics described in this section assume different Middleware home paths.
>
> Such a setup has the advantages of no PATH/ORACLE_INSTANCE conflict for services, a clean separation of web applications by JVMs, and the ability to move the Decision Manager setup to a different host.

This section contains the following topics:

- Section 1.6.7.5.1, "Creating the Domain for Oracle RTD and Decision Manager"
- Section 1.6.7.5.2, "Configuring the Oracle HTTP Server Instance"
- Section 1.6.7.5.3, "Creating the OAM Policy and the Webgate 11g Agent"
- Section 1.6.7.5.4, "Deploying the Webgate 11g Agent and Activating the Settings"

1.6.7.5.1 Creating the Domain for Oracle RTD and Decision Manager

To create the domain for Oracle RTD and Decision Manager, perform the following step:

1. Run the WebLogic domain creation wizard (from `<MWHOMEWT>\Oracle_WT1\common\bin\config.cmd` and create the domain that is going to host Oracle RTD and Decision Manager, for example, `DM_domain`.

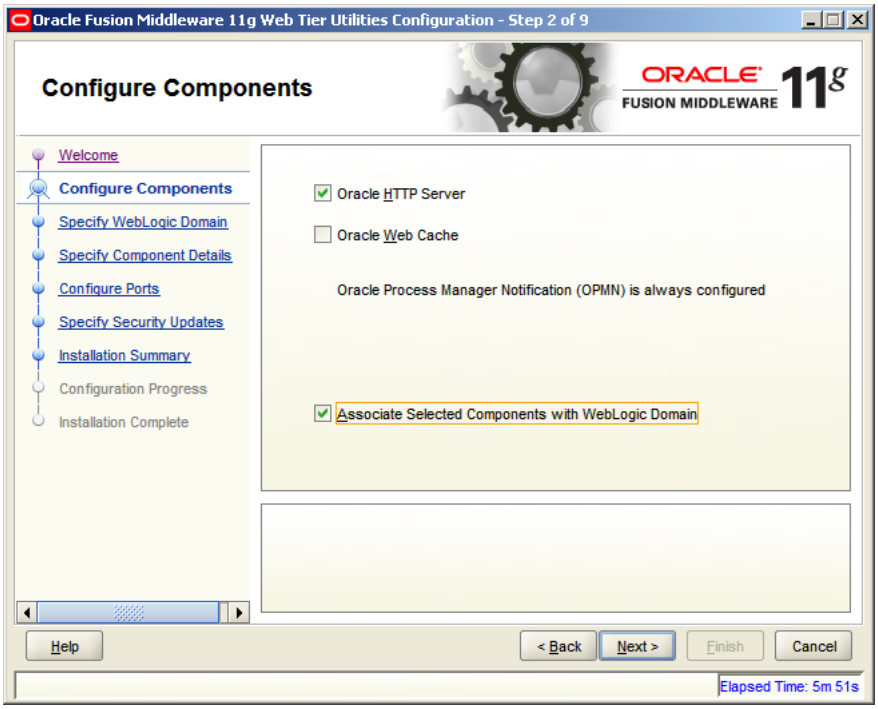
   At this point only Enterprise Manager is to be installed (which will include JRF).

   After this install, start up the AdminServer.

1.6.7.5.2 Configuring the Oracle HTTP Server Instance

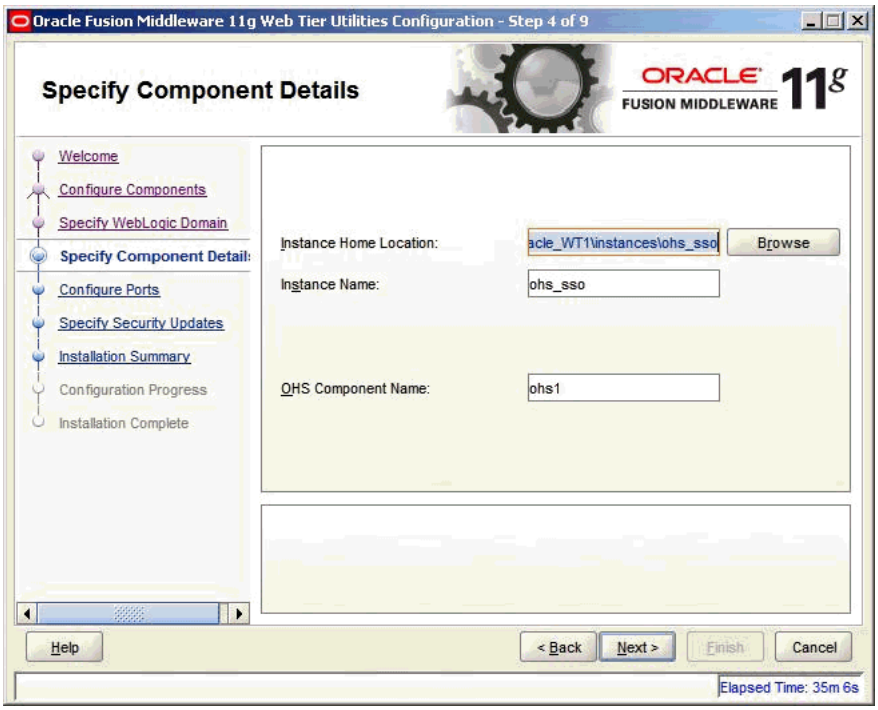
To configure the Oracle HTTP Server instance, perform the following steps:

1. Run the Web Tier instance configuration wizard (from `<MWHOMEWT>\Oracle_WT1\bin\config.bat`.
2. For the components to configure, select the following:
   - Oracle HTTP Server
   - Associate Selected Components with WebLogic Domain



> **Note:** You may also select Oracle Web Cache, to set up static resources and speed up web applications.

3. For the WebLogic domain, specify the credentials for the `DM_domain` administrator.
4. For the component details, specify the Instance Name as desired (for example `ohs_sso`), and the OHS Component Name as the standard `ohs1`.



5. For port configuration, select "auto port configuration".
6. Continue to click "Next" until the installation is complete.

1.6.7.5.3 Creating the OAM Policy and the Webgate 11g Agent

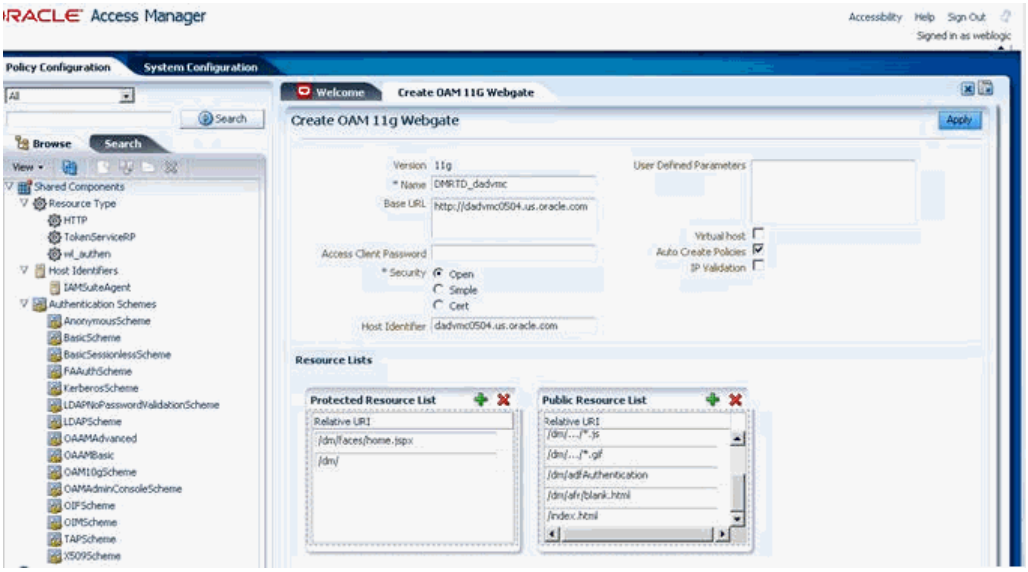
Configuring the Webgate component consists of creating the OAM Policy and the Webgate 11g agent, then deploying and activating its configuration.

To create the OAM policy and the Webgate 11g Agent, perform the following steps:

1. Open the OAM console (for example, *http://dadvmc0504.us.oracle.com:7501/oamconsole*), and log in as the admin user.
2. Click New OAM 11g Webgate and enter the Name, Base URL, and Host Identifier.

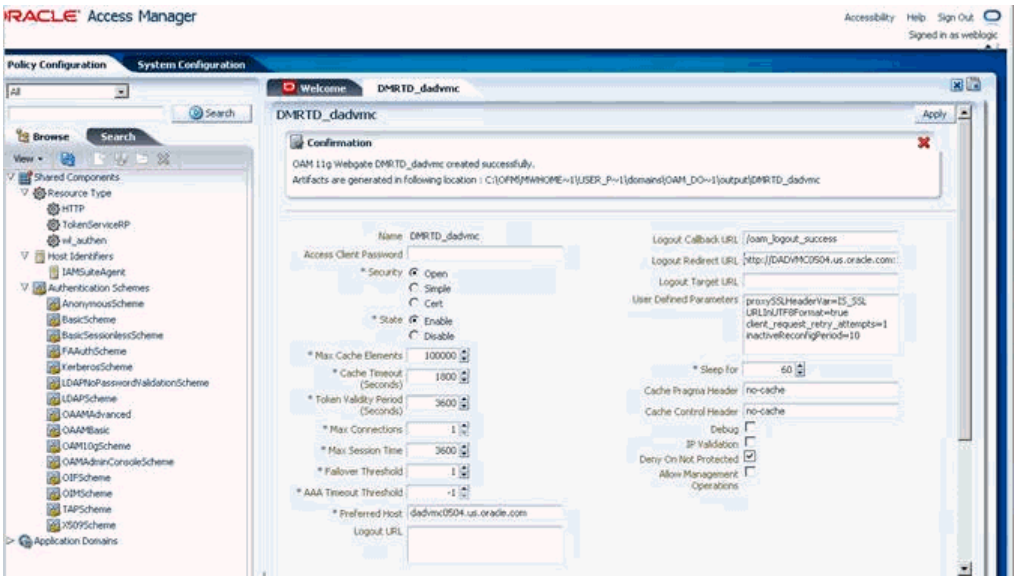
   The example shows:

   - *Name=DMRTD_dadvmc*
   - *Base URL=http://dadvmc0504.us.oracle.com*
   - *Host Identifier= dadvmc0504.us.oracle.com*



3. Edit the Host Identifier, which is set after you enter the Name.
4. For Security, select Open.
5. Specify the following Protected Resources:
   - /dm
   - /dm/
   - /dm/faces/home.jspx
   - /ui
   - /ui/
6. Specify the following Public Resources:
   - /dm/…/*.jpg
   - /dm/…/*.png
   - /dm/…/*.gif
   - /dm/…/*.css
   - /dm/…/*.js
   - /dm/adfAuthentication
   - /dm/afr/blank.html
   - /ui/…/*.jpg
   - /ui/…/*.png
   - /ui/…/*.gif
   - /ui/…/*.css
   - /ui/…/*.js
   - /ui/…/*.htc
   - /index.html
7. Click **Apply**.

This displays a screen similar to the following:



8. Close the Confirmation message
9. Edit Preferred Host, adding the port (for example, *dadvmc.us.oracle.com*:**7777**).
10. Enter the Logout URLs ***/oamsso/logout.html,/ui/do/logout***.
11. Click Apply.

    This will add port 7777 into list of recognized webhosts. Without this, :7777 will not be served by Oracle Access Manager.
12. Close the Edit Webgate page.
13. Open Host Identifiers, and double-click your new host.

> **Note:** For showing the new host under Host Identifiers, click the Refresh button.

14. Check that the new port 7777 is visible (for example, *dadvmc.us.oracle.com*:**7777)**. If not, add a new line.
15. After editing the policy, restart the OAM server *(this is necessary to make the policy work)*.

1.6.7.5.4 Deploying the Webgate 11g Agent and Activating the Settings

> **Note:** This section describes the steps for deploying the Webgate 11g Agent for Windows. To see how to configure the Webgate 11g Agent on Linux, see the Section "Post-Installation Steps" in *Oracle Fusion Middleware Installation Guide for Oracle Identity Management*.

To deploy the Webgate 11g agent and activate the settings, perform the following steps:

1. Stop the OHS server if it is running.
2. Deploy the provisioned agent configuration into the actual location:

```
set MWHOME= C:\OFM\MWHOMEWT
cd %MWHOME%/Oracle_OAMWebGate1/webgate/ohs/tools/deployWebGate
deployWebGateInstance.bat -w %MWHOME%\Oracle_WT1\instances\ohs_
sso\config\OHS\ohs1 -oh %MWHOME%\ORACLE_OAMWebGate1
```

This should return results similar to the following:

```
Copying files
C:\OFM\MWHOMEWT\Oracle_OAMWebGate1\webgate\ohs\config\oblog_config_wg.xml
1 File(s) copied
C:\OFM\MWHOMEWT\Oracle_
OAMWebGate1\webgate\ohs\tools\openssl\simpleCA\cacert.pem
1 File(s) copied
C:\OFM\MWHOMEWT\Oracle_OAMWebGate1\webgate\ohs\tools\openssl\simpleCA\cakey.pem
1 File(s) copied
```

3. Continue by entering the following:

```
cd %MWHOME%\Oracle_OAMWebGate1\webgate\ohs\tools\EditHttpConf
set PATH=%MWHOME%\Oracle_WT1\bin; %MWHOME%\Oracle_WT1\lib; %PATH%
EditHttpConf -w %MWHOME%\Oracle_WT1\instances\ohs_sso\config\OHS\ohs1 -oh
%MWHOME%\Oracle_OAMWebGate1 -o Edithttpconf.log
```

This should return results similar to the following:

```
The web server configuration file was successfully updated
%MWHOME%\Oracle_WT1\instances\ohs_sso\config\OHS\ohs1/httpd.conf
has been backed up as %MWHOME%\Oracle_WT1\instances\ohs_
sso\config\OHS\ohs1/httpd.conf.ORIG
```

4. Edit the WebLogicHost and WebLogic port parameters in the file `%MWHOME%\Oracle_WT1\instances\ohs_sso\config\OHS\ohs1\mod_wl_ohs.conf`, as in the following example:

```
<IfModule weblogic_module>
      WebLogicHost dadvmc0504.us.oracle.com
      WebLogicPort 7401
      Debug ERR
      WLLogFile <specify MWHOME>\Oracle_WT1\instances\ohs_
sso\diagnostics\logs\OHS\ohs1\weblogic.log
</IfModule>

<Location /dm>
      SetHandler weblogic-handler
</Location>

<Location /ui>
      SetHandler weblogic-handler
</Location>
```

Note: 1. Ensure that the WebLogic server listens on the IPv4 stack. If the default network stack is IPv6, then you must edit the file `<MWHOMEWT>\user_projects\domains\DM_domain\bin\setDomainEnv.cmd` as follows:

Add

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Djava.net.preferIPv4Stack=true
```

before

```
if "%PRODUCTION_MODE%"=="true" (
```

This ensures that the WebLogic server listens on the IPv4 stack only and enables Oracle RTD to function correctly.

2. If your operating system has IPv6 as the default network stack, specify a numeric IP address, such as `WebLogicHost 127.0.0.1`, for OHS to open. Do *not* specify localhost or host name, otherwise the IPv6 stack will be attempted, and Webgate does not work correctly with IPv6.

5. Copy

```
%MWHOME%/Oracle_WT1/instances/ohs_sso/config/OHS/ohs1/
webgate/config/oblog_config_wg.xml
```

into

```
%MWHOME%/Oracle_OAMWebGate1/webgate/ohs/config/oblog_
config.xml
```

6. Start OHS, as follows:

```
cd %MWHOME%/Oracle_WT1/instances/ohs_sso/bin
opmnctl status
```

After the message appears that `opmn` is not running, continue by entering:

```
opmnctl start
opmnctl status
```

The status details appear, as in the following example:

```
Processes in Instance: ohs_sso
---------------------------------+--------------------+---------+---------
ias-component                    | process-type       |     pid | status
---------------------------------+--------------------+---------+---------
ohs1                             | OHS                |   28262 | Alive
```

1.6.7.6 Configuring Domain for Oracle RTD and Decision Manager

This section contains the following topics:

- Section 1.6.7.6.1, "Configuring a New Authenticator for Oracle WebLogic Server"
- Section 1.6.7.6.2, "Configuring OAM as New Identity Asserter for Oracle WebLogic Server"
- Section 1.6.7.6.3, "Enabling SSO Authentication Using Fusion Middleware Control"
- Section 1.6.7.6.4, "Deploying the Applications"

1.6.7.6.1 Configuring a New Authenticator for Oracle WebLogic Server

After installing Oracle RTD Decision Management, the Oracle WebLogic Server embedded LDAP server is the default authentication source (identity store). To use a new identity store (for example, OID), as the main authentication source, you must configure the Oracle WebLogic Server domain (where Oracle RTD and Oracle RTD Decision Management have been installed).

For more information about configuring authentication providers in Oracle WebLogic Server, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

To configure a new authenticator in Oracle WebLogic Server:

1. Log in to Oracle WebLogic Server Administration Console and click **Lock & Edit** in the Change Center.
2. Select **Security Realms** from the left pane and click **myrealm**.

   The default Security Realm is named **myrealm**.
3. Display the **Providers** tab, then display the **Authentication** sub-tab.
4. Click **New** to launch the **Create a New Authentication Provider** page.

   Complete the fields as follows:

   - **Name**: *OID Provider*, or a name of your choosing.
   - **Type**: OracleInternetDirectoryAuthenticator
   - Click **OK** to save the changes and display the authentication providers list updated with the new authentication provider.
5. Click the newly added authenticator in the **authentication providers** table.
6. Navigate to **Settings**, then select the **Configuration\Common** tab:
   - Select **SUFFICIENT** from the **Control Flag** list.
   - Click **Save**.
7. Display the **Provider Specific** tab and specify the following settings using appropriate values for your environment:

Section Name	Field Name	Description
Connection	Host	The LDAP host name. For example, *<localhost>*.
Connection	Port	The LDAP host listening port number. For example, 6050.
Connection	Principal	The distinguished name (DN) of the user that connects to the LDAP server. For example, cn=orcladmin.
Connection	Credential	The password for the LDAP administrative user entered as the Principal.
Users	User Base DN	The base distinguished name (DN) of the LDAP server tree that contains users. For example, use the same value as in Oracle Access Manager.
Users	All Users Filter	The LDAP search filter. For example, (&(uid=*) (objectclass=person)). The asterix (*) filters for all users. Click More Info... for details.

Section Name	Field Name	Description
Users	User From Name Filter	The LDAP search filter. Click More Info... for details.
Users	User Name Attribute	The attribute that you want to use to authenticate (for example, cn, uid, or mail). Set as the default attribute for user name in the directory server. For example, *uid*. **Note**: The value that you specify here must match the User Name Attribute that you are using in the authentication provider, as described in as described in "Configuring the User Name Attribute in the Identity Store" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
Groups	Group Base DN	The base distinguished name (DN) of the LDAP server tree that contains groups (same as User Base DN).
General	GUID attribute	The attribute used to define object GUIDs in LDAP. orclguid **Note**: You should not normally change this default value, however, if you do, you must also specify the changed value in Fusion Middleware Control, as described in "Configuring the GUID Attribute in the Identity Store" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.

For more information about configuring authentication providers in Oracle WebLogic Server, see *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

8. Click **Save**.
9. Perform the following steps to set up the default authenticator for use with the Identity Asserter:
   a. At the main **Settings for myrealm** page, display the **Providers** tab, then display the **Authentication** sub-tab, then select **DefaultAuthenticator** to display its configuration page.
   b. Display the **Configuration\Common** tab and select 'SUFFICIENT" from the **Control Flag** list.

      For more information, see "Setting the JAAS Control Flag Option" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
   c. Click **Save**.
10. Perform the following steps to reorder Providers:
    a. Display the **Providers** tab.
    b. Click **Reorder** to display the **Reorder Authentication Providers** page

c. Select a provider name and use the arrow buttons to order the list of providers as follows:
   - OID Authenticator (SUFFICIENT)
   - OAM Identity Asserter (REQUIRED)
   - Default Authenticator (SUFFICIENT)

d. Click **OK** to save your changes.

11. In the Change Center, click **Activate Changes**.
12. Restart Oracle WebLogic Server.

1.6.7.6.2 Configuring OAM as New Identity Asserter for Oracle WebLogic Server

The Oracle WebLogic Server domain in which Oracle RTD and Oracle RTD Decision Management have been installed must be configured to use an Oracle Access Manager asserter.

For more information about creating a new asserter in Oracle WebLogic Server, see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

To configure Oracle Access Manager as the new asserter for Oracle WebLogic Server:

1. Log in to Oracle WebLogic Server Administration Console.
2. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**. Select **Providers**.
3. Click **New**. Complete the fields as follows:
   - **Name**: *OAM Provider*, or a name of your choosing.
   - **Type**: OAMIdentityAsserter.
4. Click **OK**.
5. Click **Save**.
6. In the **Providers** tab, perform the following steps to reorder **Providers**:
   a. Click **Reorder**
   b. In the **Reorder Authentication Providers** page, select a provider name, and use the arrows beside the list to order the providers as follows:
      - OID Authenticator (SUFFICIENT)
      - OAM Identity Asserter (REQUIRED)
      - Default Authenticator (SUFFICIENT)
   c. Click **OK** to save your changes.
7. In the Change Center, click **Activate Changes**.
8. Restart Oracle WebLogic Server.

   You can verify that Oracle Internet Directory is the new identity store (default authenticator) by logging back into Oracle WebLogic Server and verifying the users and groups stored in the LDAP server appear in the console.
9. Use Fusion Middleware Control to enable SSO authentication.

   For more information, see Section 1.6.7.6.3, "Enabling SSO Authentication Using Fusion Middleware Control".

1.6.7.6.3 Enabling SSO Authentication Using Fusion Middleware Control

After Oracle RTD Decision Management has been configured to use the SSO solution configured for use by Oracle Fusion Middleware, you must enable SSO authentication for Oracle RTD Decision Management in Fusion Middleware Control from the **Security** tab.

To enable Oracle RTD Decision Management to use SSO authentication:

1. Log in to Fusion Middleware Control.
2. Go to the **Security** page and display the **SSO** tab.

   Click the **Help** button on the page to access the page-level help for its elements.
3. Click **Lock and Edit Configuration**.
4. Select **Enable SSO**.

   When selected, this checkbox enables SSO to be the method of authentication into Oracle RTD Decision Management. The appropriate form of SSO is determined by the configuration settings made for the chosen SSO provider.
5. Select the configured SSO provider from the list.

   The SSO provider list becomes active when you select the **Enable SSO** checkbox.
6. If required, enter logon and logoff URLs for the configured SSO provider.

   The logoff URL (specified by the SSO provider) must be outside the domain and port that the SSO provider protects, because the system does not log users out.
7. Click **Apply**, then **Activate Changes**.
8. Restart the Oracle RTD Decision Management components using Fusion Middleware Control.

1.6.7.6.4 Deploying the Applications

Perform the following steps:

1. Start the DM_domain.
2. Prepare OracleRTD.ear and clm.ear, as follows:
   - Unpack each .ear file into a separate directory.
   - Edit web.xml, replacing **<auth-method>FORM</auth-method>** with **<auth-method>CLIENT-CERT</auth-method>**
   - Re-pack the .ear files with the changed web.xml file.
3. As for standard Oracle RTD and Oracle RTD Decision Management installations *(for details, see previous sections of this manual and Oracle Real-Time Decisions Installation and Administration Guide)*, create the data sources, and install the .ear files.

   Note: You must create your users in LDAP and assign them to appropriate groups. For details, see Section 1.6.7.4, "Creating Groups and Users in LDAP Server."
4. As a final step, verify that the Decision Manager URL login redirects as configured, and that you can log in to the Decision Manager correctly.

   For example, depending on your ports, opening

http://*<web_tier_host>*:7777/dm/

in a browser should redirect into a URL similar to

http://*<oam_server_host>*:14100/oam/server/obrareq.cgi?encquery%..

Then, log in to the Decision Manager with a correct user (for example, clmuser) to complete the verification.

1.6.7.7 Troubleshooting

You may encounter one or more of the issues in this section.

1. If OS has IPv6 protocol used by default, not IPv4, then webgate log files will have numerous error messages

For example,

AUTHENTICATION ERROR 0x00001823 ..\src\obuser_session.cpp:1939 "Location in an ObUserSession constructor is not a valid host."

In order to stop this spam from collecting, edit the file

C:\OFM\MWHOMEWT\Oracle_WT1\instances\ohs_sso\config\OPMN\opmn\opmn.xml

to specify

<notification-server interface="***ipv4***">

2. Apache access log file has a new record for OPMN ping query for every 20 seconds

Edit the file:

C:\OFM\MWHOMEWT\Oracle_WT1\instances\ohs_sso\config\OPMN\opmn\opmn.xml

Insert the following *<ping interval...>* line into the *<process-type...>* section, as in the following example

```
  <process-type id="OHS" module-id="OHS1">
    <module-data>
      <category id="start-parameters">
        <data id="start-mode" value="ssl-enabled"/>
      </category>
    </module-data>
    <ping interval="180"/>
    <process-set id="OHS" numprocs="1"/>
  </process-type>
</ias-component></ias-instance>
```

3. URL /dm results in error 404

Into the file C:\OFM\MWHOMEWT\Oracle_WT1\instances\ohs_sso\config\OHS\ohs1\httpd.conf, insert the following lines:

```
RewriteEngine on
RewriteRule ^/dm$ /dm/ [R]
RewriteRule ^/dm\?(.*)$ /dm/?$1 [R]
```

Restart the OHS server.

If you now open

`http://<web_tier_host>:7777/dm`

this will get redirected to

`http://<web_tier_host>:7777/dm/`

1.6.8 Using SSO with Oracle RTD Decision Management Applications on WebSphere

This section describes how to set up single sign-on (SSO) solution with Oracle RTD Decision Management on WebSphere.

This section contains the following topics:

1.6.8.1 Configuring DMCOOKIE

To configure DMCOOKIE, perform the following steps:

1. Login to the console.
2. In the tree on the left, expand Applications, and then choose Application types, then WebSphere enterprise applications.
3. Click OracleRTD.
4. Under Modules, click Manage Modules.
5. Click Oracle RTD Decisions Manager.
6. Under Additional Properties, click Session Management.
7. Under Session tracking mechanism, ensure that Enable cookies is checked and click Enable cookies.
8. Set Cookie name to DMCOOKIE.
9. Set Cookie path to /dm.
10. Click OK, then click Save.

1.6.8.2 Configuring UICOOKIE

To configure UICOOKIE, perform the following steps:

1. Login to the console.
2. In the tree on the left, expand Applications, and then choose Application types, then WebSphere enterprise applications.
3. Click OracleRTD.
4. Under Modules, click Manage Modules.
5. Click Oracle RTD decision center.
6. Under Additional Properties, click Session Management.
7. Under Session tracking mechanism, ensure that Enable cookies is checked and click Enable cookies.
8. Set Cookie name to UICOOKIE.

9. Set Cookie path to /ui.
10. Click OK, then click Save.

1.6.8.3 Configuring SOAPCOOKIE

To configure SOAPCOOKIE, perform the following steps:

1. Login to the console.
2. In the tree on the left, expand Applications, and then choose Application types, then WebSphere enterprise applications.
3. Click OracleRTD.
4. Under Modules, click Manage Modules.
5. Click Oracle RTD soap.
6. Under Additional Properties, click Session Management.
7. Under Session tracking mechanism, ensure that Enable cookies is checked and click Enable cookies.
8. Set Cookie name to SOAPCOOKIE.
9. Set Cookie path to /soap.
10. Click OK, then click Save.

> **Note:** After you have configured all the cookies and without LTPA is enabled, the Oracle RTD Decision Center login screen appears after you log into the Decision Manager web application.

1.6.8.4 Enabling LTPA

To enable LTPA, perform the following steps:

1. Login to the console.
2. In the tree on the left, expand Security, and then choose Global security.
3. Under Authentication, ensure that LTPA is selected, and click LTPA.
4. Expand Web and SIP security and click Single sign-on (SSO).
5. Select Enabled.
6. Click OK, then click Save.
7. Restart WebSphere.

2

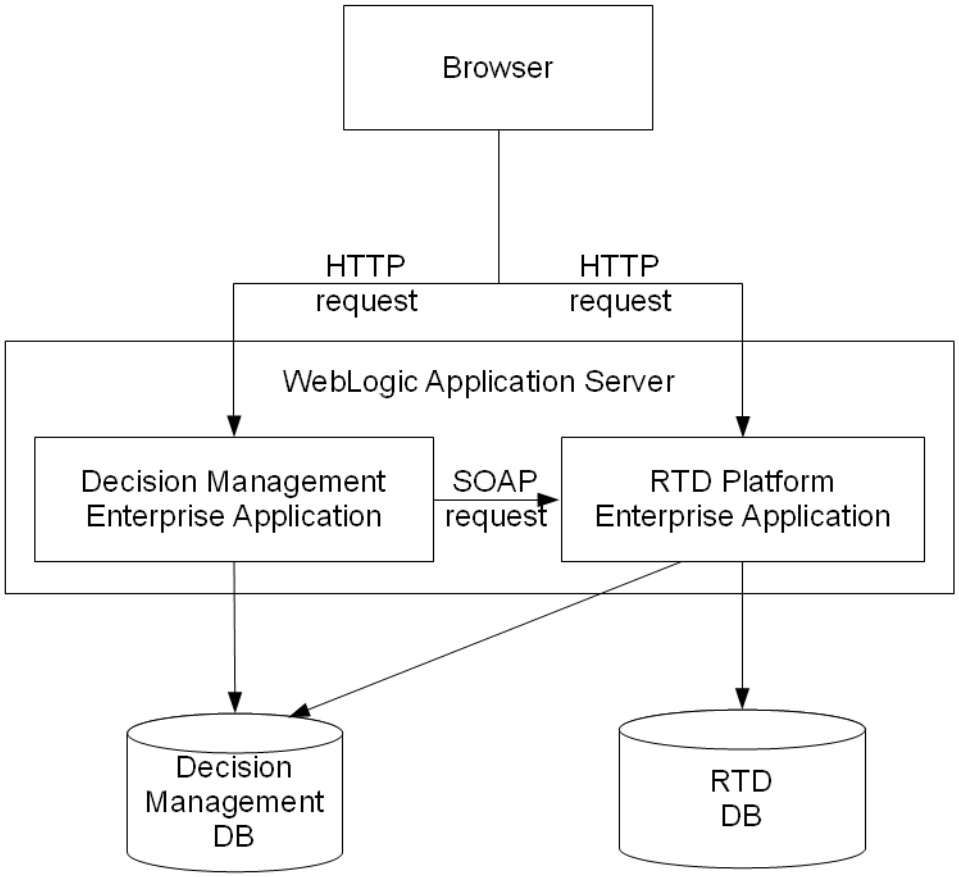
Configuring Oracle RTD Decision Management

Terminology: The term "reference implementation" is used in this chapter to refer to the specific Oracle RTD Decision Management application RTD for Marketing Optimization, released with Oracle RTD Base Application.

This chapter contains the following topics:

2.1 Oracle RTD Decision Management Architecture Overview

The following diagram shows the communications at runtime between the various components of Oracle RTD Decision Management after installation:



The Oracle RTD Decision Management application is an ADF based web application, which also leverages the Oracle RTD reports as JSP pages embedded in an IFrame. If Decision Center reports are enabled and single sign-on configured, the Decision Manager browser displays JSP pages from both the Oracle RTD Decision Management and the Oracle RTD applications.

Additionally, Oracle RTD makes web service (SOAP) calls from the Oracle RTD Decision Management application to the Oracle RTD workbench server to retrieve type restrictions and display the rule editor. These requests use the username and password that you configured in Section 1.4.3.5, "Storing Credentials to Enable Web Service Calls" for WebLogic or Section 1.5.2.5, "Storing Credentials to Enable Web Service Calls" for WebSphere.

The Oracle RTD Decision Management application accesses the Decision Management database using the credentials defined in the CLMDS JNDI datasource.

The Oracle RTD Decision Management Inline Service loaded in the Oracle RTD application accesses the Decision Management database to retrieve dynamic choices.

The Oracle RTD application accesses the Oracle RTD (SDDS) database using the credentials defined in the SDDS JNDI datasource.

2.2 Configuring Oracle RTD Decision Management

This section contains the following topics:

2.2.1 Overview

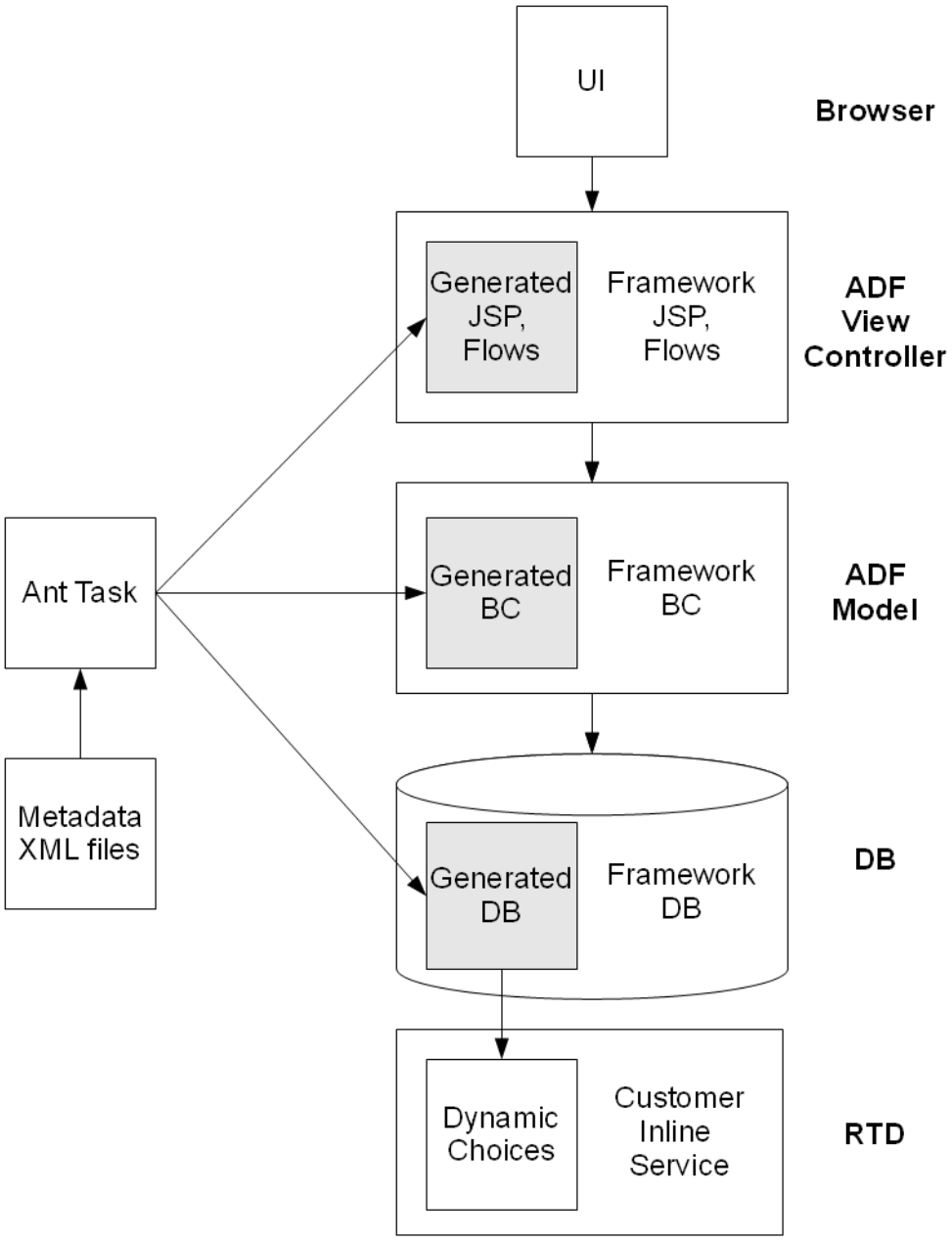
Oracle RTD Decision Management applications are built and deployed using Decision Designer.

Decision Designer is the combination of design tools that include JDeveloper, ant, and the metadata files that you set up and used to create the Oracle RTD Decision Management application, as described in Chapter 1, "Installing Oracle RTD Decision Management."

You can use Decision Designer to make changes to the reference implementation, or to build your own application from scratch.

You can also use Decision Designer to further customize the behavior of Oracle RTD Decision Management applications by adding your own logic. You can do so in the model layer and the view controller layer. The two ways to achieve customization are through extensibility of the ADF Framework and through the Decision Manager templates. As an example, the rule to validate that a start date is less than an end date has been implemented through templates in the released application RTD for Marketing Optimization

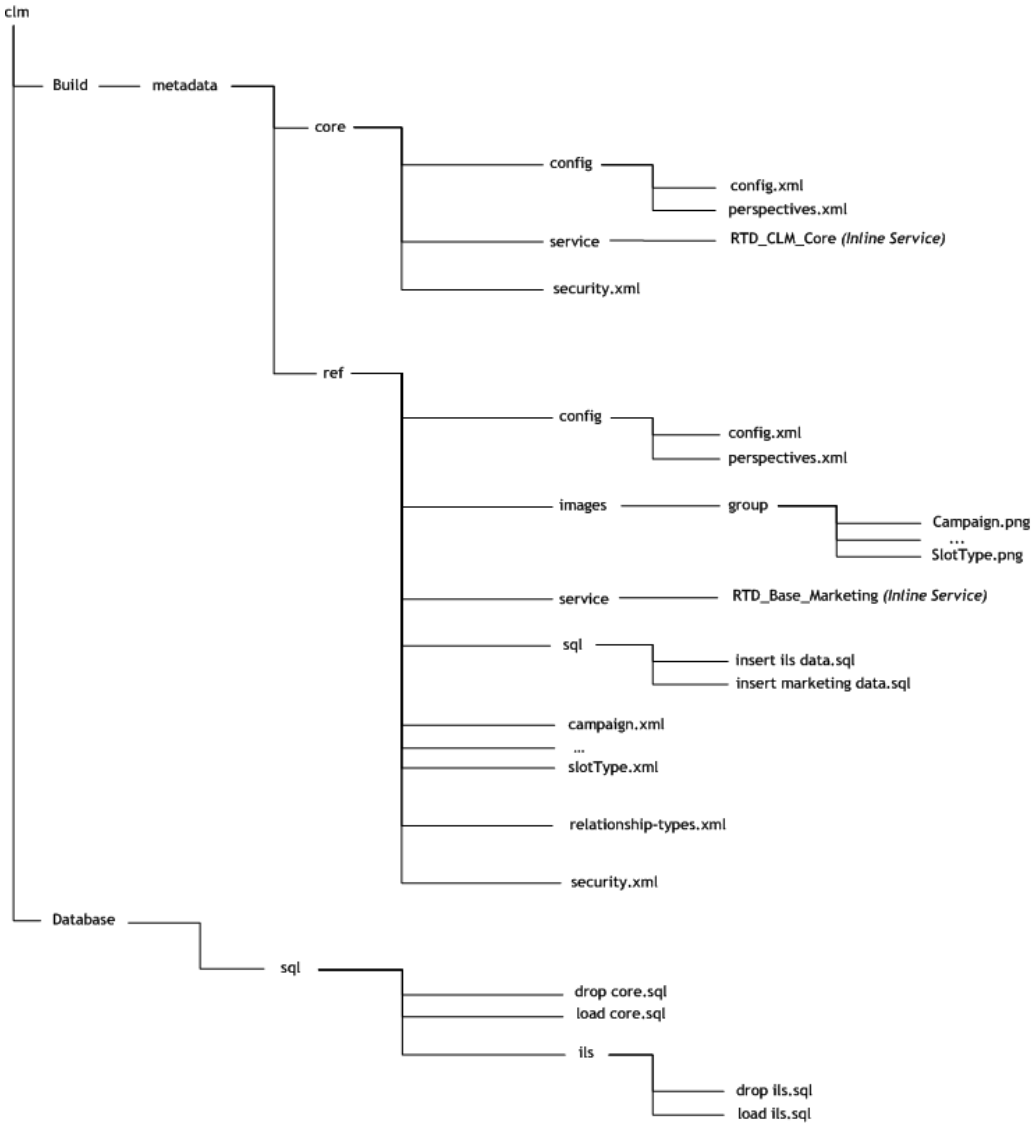
The following diagram shows the components that are generated by Decision Designer.



2.2.2 Oracle RTD Decision Management Metadata Configuration Files

The following diagram shows the main Decision Management metadata and database setup files, as released with Oracle RTD Base Application.

Figure 2–1 Main Oracle RTD Decision Management Metadata and Database Files



The folder **clm\Build\metadata**, also referred to generically as *<metadata_modules_home>*, contains metadata modules for different applications.

The folder **clm\Build\metadata\ref** contains the metadata for the reference implementation.

Within a metadata module, two files, `config.xml` and `perspectives.xml`, must be in the **config** subdirectory. And you can add as many files as you want in the main directory to configure choice groups, relationship types, and security.

The folder **clm\Build\metadata\core** contains the metadata module for the core application. Use the core application when you want to create your own application from scratch, it contains the strict minimum for an empty application and Inline Service.

Note: In all the XML files, the characters "<" and ">" are effectively control characters, and must not be used to specify any metadata element values. For operators, for example, they must be replaced by "<"(instead of "<") and ">"(instead of ">").

This section contains the following topics:

- Section 2.2.2.1, "Config XML File"
- Section 2.2.2.2, "Perspectives XML File"
- Section 2.2.2.3, "Choice Group, Project, and Relationship-Types XML Files"
- Section 2.2.2.4, "User Interface Pages"
- Section 2.2.2.5, "Security XML Files"
- Section 2.2.2.6, "Folder for Choice Group Images"
- Section 2.2.2.7, "Inline Service Folder"

2.2.2.1 Config XML File

`config.xml` contains general configuration settings:

Under a <config> node, the following settings are available:

- <workbenchService> specifies the url of the Oracle RTD server running in the same host WebLogic server as Oracle RTD Decision Management. The url must contain the protocol (http or https), the host, and the port.

  The host is therefore always **localhost**, but the port can change based on your configuration. If you followed the instructions in Chapter 1, "Installing Oracle RTD Decision Management," the default port is 7001 in a development WebLogic environment, 7003 in a production WebLogic environment, and 9080 in a WebSphere environment.

- <inlineService> specifies the name of the Inline Service used to get type restrictions and Decision Center reports.

  The deployment state must be the exact deployment state that you want to access, 5 by default.

- <decisionCenter> specifies the base URL to access embedded Decision Center reports. Remove the <decisionCenter> node to hide the Decision Center reports in the user interface.

- <ownership-mode> specifies the ownership mode for when a choice is edited in a project. The value can be:

  – off (the default)

    The user does not become the owner of the choice when the user edits a choice, and the choice owner is not shown in the Decision Manager user interface.

  – on

    The user becomes the owner of the choice when the user edits a choice, and the choice owner is shown in the Decision Manager user interface. Users can also see the owner in the Audit Trail without adding the Owner attribute on the page.

- <display-name> specifies the name of the application as it appears on the login page and at the top of the main page.
- <database-encoding>

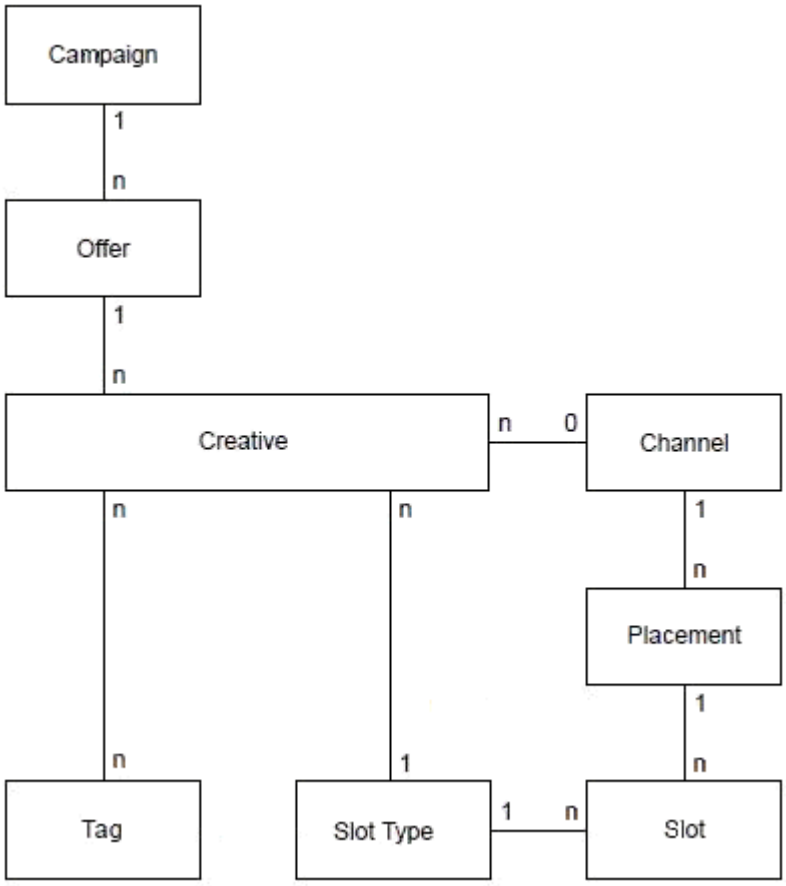
  The value of <database-encoding> depends on the database character set. Oracle RTD Decision Management uses database encoding to properly enforce byte length throughout the application.

  You must specify the java character set, as defined in java.nio.charset.Charset. For example, for an Oracle database with charset AL32UTF8, the value of <database-encoding> must be UTF-8. UTF-8 is the default value, used when <database-encoding> is not specified in config.xml.
- <enterpriseAppName> specifies the application name used when multiple applications are deployed into the same domain. The default value is clm.
- <datasourceJNDIName> specifies the data source name used when multiple applications are deployed into the same domain. The default value is CLMDS.
- <webAppContextRoot> specifies the context root used when multiple applications are deployed into the same domain. This element must start with "/", for example "/dm2". The default value is /dm.

2.2.2.2 Perspectives XML File

Perspectives allow business users to view a subset of the choice group graph in the folder tree in the Decision Manager user interface.

For instance, the reference implementation contains the choice groups shown in the following choice group graph:



Four "general" - that is, unqualified - perspectives are defined on top of this graph:

Figure 2–2 Campaigns Perspective

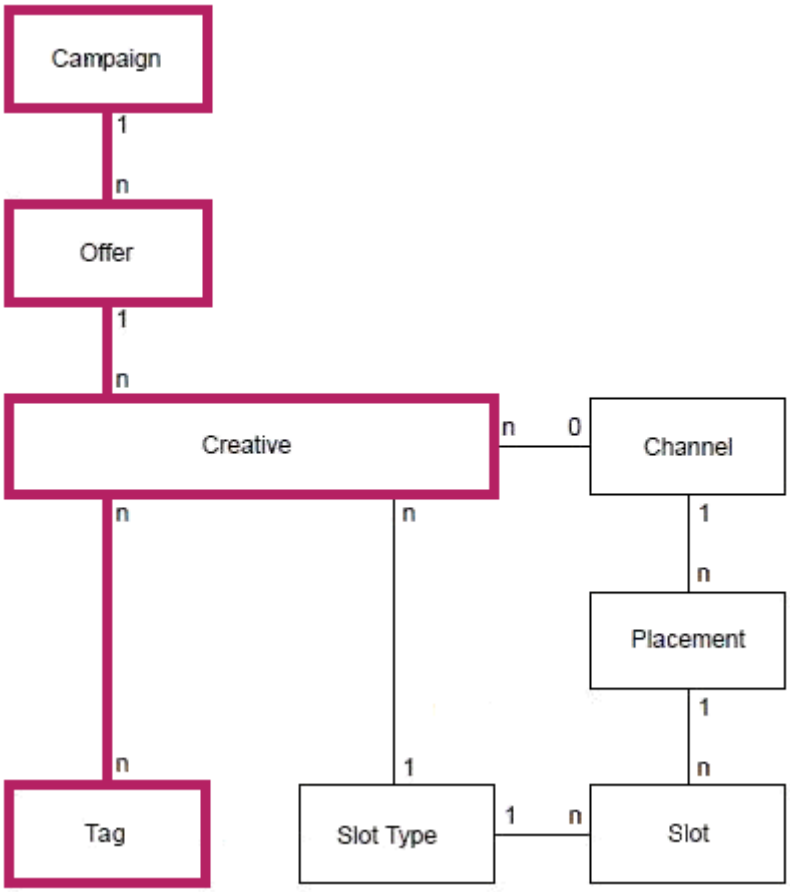


Figure 2–3 Tags Perspective

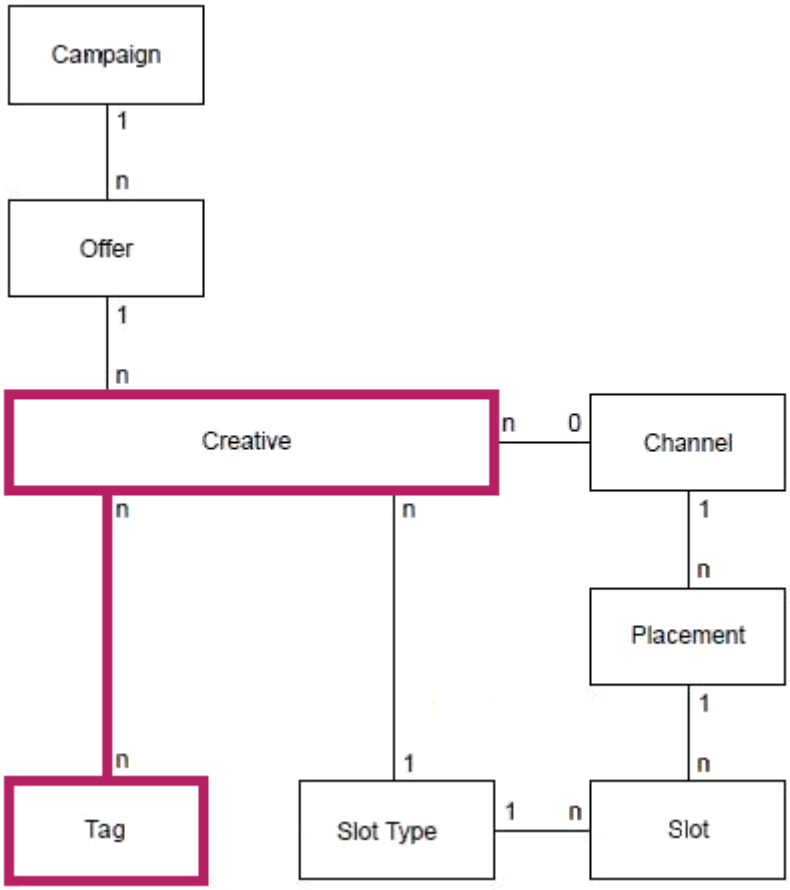


Figure 2–4 Slot Types Perspective

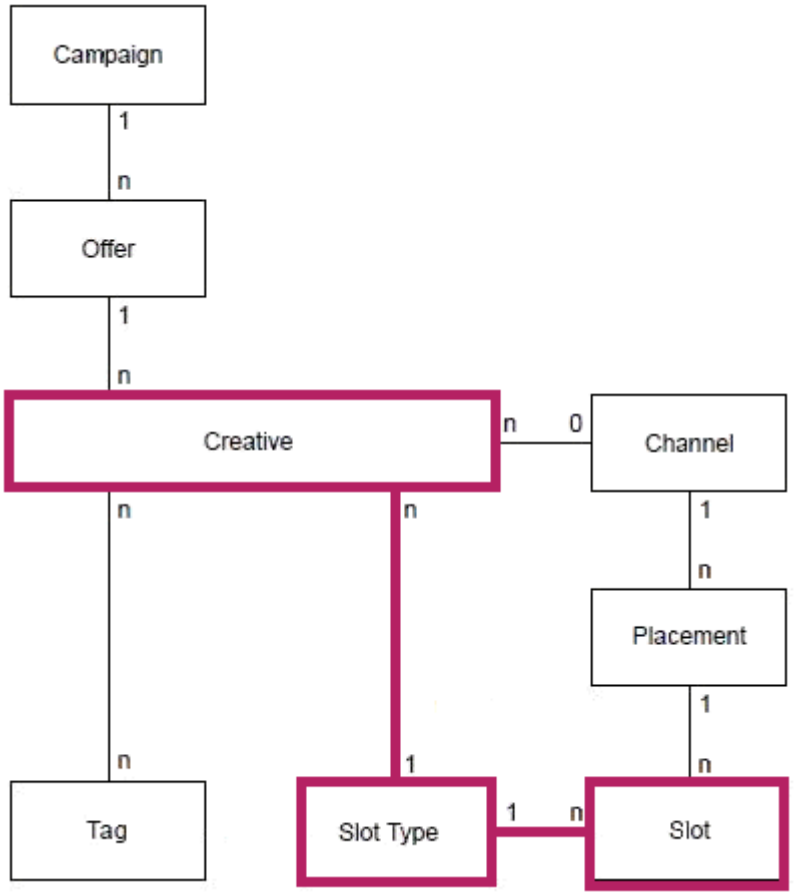
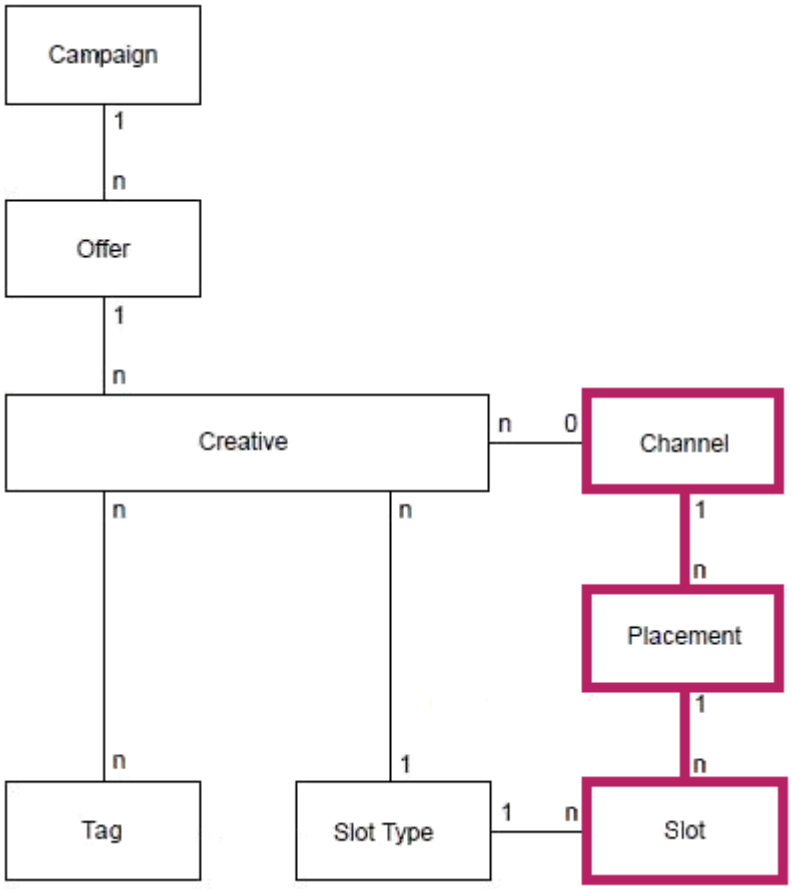


Figure 2–5 Channels Perspective



Perspectives are defined in the file `clm\Build\metadata\ref\config\perspectives.xml` (you can modify this if you use a metatada.module different from ref).

`perspectives.xml` is made of a <perspectives> root node which contains multiple perspectives.

> **Note:** In this section, property entries shown in bold are required entries.

Properties of perspective:

- **name**: the name of the perspective
- description

  If present in perspectives.xml, the description appears as the heading for the perspective in the left-hand Perspective panel of the Decision Manager user interface. It also provides the tool tip help information for Decision Manager users as they mouse hover over the perspective name either in the Perspective panel or the dropdown list of perspectives that is displayed by clicking the Perspectives list icon at the top of the Perspectives panel.
- one **root** node
- multiple level nodes (ordered)

A single root node can have multiple level nodes in sequence - which enables multiple folders to be visible in sequence under the root choice group in the Decision Manager user interface. Alternatively, to enable a hierarchical folder structure in the Decision Manager user interface, the level nodes under a root node can be nested.

For example, the associations Slot Type -> Creative, Slot-Type -> Slot can be represented through the following schematic:

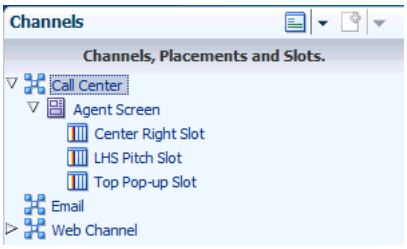
```
<root ... "Slot Type"... >
   <level ... "Creative" ... />
   <level ... "Slot" ... />
</root>
```

In the Decision Manager user interface, this provides a perspective panel display similar to the following (with showFolder=true):

On the other hand, a hierarchical association Channel -> Placement -> Slot can be defined with nested level nodes, as in the following schematic:

```
<root ... "Channel" ... >
   <level ... "Placement" ... >
      <level ... "Slot" ... />
   </level>
</root>
```

In the Decision Manager user interface, this provides a perspective panel display similar to the following:



Properties of root:

- **choiceGroupId**: the choice group that will be the root node of the tree
- simpleViewCriteria (see later in this section)
- viewCriteria (see later in this section)

Properties of level:

- **choiceGroupId**: the choice group that will be the node of the tree at that level
- **relationshipTypeId**: the relationship to follow between the previous level and this level

- reversed: whether to follow the relationship from fromChoiceGroupId to toChoiceGroupId (reversed = false) or from toChoiceGroupId to fromChoiceGroupId (reversed = true, the default).

  For the cardinalities *:1, *:0, *:*, and *:1..*, use reversed=true to display many child nodes under each node

- showFolder

  If showFolder=true, this shows a folder with the name of the relationship (when reversed=true, relationship types folder name is toName, when reversed=false, relationship types folder name is fromName)

  If showFolder=false (the default value), this shows the children directly under the parent node without a folder.

- simpleViewCriteria (see later in this section)
- viewCriteria (see later in this section)

simpleViewCriteria and viewCriteria allow the application of a filter on which choices are displayed. You can specify multiple choices and the resulting filter will be the OR combination of all of them (that is, a choice will appear if it matches any of the criteria).

Properties of simpleViewCriteria:

- attribute: the choice attribute to filter on (of the choice group defined by choiceGroupId for this root or level)
- operator: the operator to use for filtering (see searchCriteriaOperator later in this section for a list of operators)
- value: the value of the choice attribute to filter on

Example:

```
<perspective name="Draft Campaigns"> ......
<root choiceGroupId="Campaign">
<simpleViewCriteria attribute="approvalStatus" value="Draft"/>
```

Note: To specify criteria for date attributes, the value must be in the format yyyy-mm-dd, such as in:

```
<simpleViewCriteria attribute="startDate" operator="AFTER"
value="2012-11-25" />
```

Properties of viewCriteria:

- name: the id of a criteria defined on the choice group of the same choiceGroupId as this root or level

2.2.2.3 Choice Group, Project, and Relationship-Types XML Files

You can put as many xml configuration files in the top-level application configuration folder (*for the reference implementation, this is* **clm\Build\metadata\ref**) as you wish.

These configuration files let you specify choice groups and relationship types between choice groups.

Relationship types are owned by one side. In other words, to add, edit or delete a relationship between two choices, the user must be able to lock the choice owning that relationship.

When defining choice groups, you also define all the attributes in each choice group and all the user interface views associated with displaying this choice group in the Decision Manager user interface.

When defining choice attributes, you can define which widget is used to display this attribute. Oracle RTD Decision Management automatically selects a default widget most appropriate for the attribute, but you can override that and specify a different widget.

For the purpose of defining objects and their attributes, projects are considered as choice groups in metadata. The only difference is that you cannot have relationship types between projects and other choice groups. You can define project attributes and project views similar to the way you describe choice group attributes and views. You can even have project attributes with type restrictions by having a choice group in your Inline Service with id Project.

Note: In the following lists, properties shown in bold mean that they are required.

Properties of choice groups:

- **id**: the id of the choice group. It must match the id in the Inline Service
- **name**: the name of the choice group. It will appear as such in the Decision Manager user interface
- description: the description of the choice group.
- searchSortOrder: the order in which this choice group will appear in the search dropdowns (sorted ascending based on this number)

  If you do not put this property, then this choice group will not appear in the dropdowns
- createSortOrder: the order in which this choice group will appear in the create dropdown (sorted ascending based on this number)

  If you do not put this property, then this choice group will not appear in the dropdown
- attributes: an array of attribute (see later in this section)
- criteria: zero to many criteria used for perspective filtering (see later in this section)

Properties of attributes:

- **id**: the id of the choice attribute. It must match the id in the Inline Service.
- **name**: the name of the choice attribute. It will appear as such in the Decision Manager user interface.
- description: the description of the choice attribute.This description will appear in tool tips in the user interface, and is a useful way to give advice to business users on how to enter values for this attribute.
- searchCriteriaOperator: the operator used for the advanced search.

  If you do not put this property, then this attribute will not show up in the advanced search (unless you use the Add Field dropdown).

  Valid values for strings:

  – "<>" (Not equal to)

  – "NOTBETWEEN" (Not between)

  – "CONTAINS" (Contains)

- "ISNOTBLANK" (Is not blank)
- "=" (Equal to)
- "<" (Less than)
- ">" (Greater than),
- "ISBLANK" (Is blank)
- "<=" (Less than or equal to)
- ">=" (Greater than or equal to)
- "DOESNOTCONTAIN" (Does not contain)
- "STARTSWITH" (Starts with),
- "ENDSWITH" (End with)
- "BETWEEN" (Between)

Valid values for dates:

- "<>" (Not equal to)
- "ISBLANK" (Is blank)
- "ONORAFTER" (On or after)
- "BEFORE" (Before)
- "NOTBETWEEN" (Not between)
- "ISNOTBLANK" (Is not blank)
- "ONORBEFORE" (On or before)
- "AFTER" (After)
- "BETWEEN" (Between)
- "=" (Equal to)

Valid values for numbers:

- "<>" (Not equal to)
- "ISBLANK" (Is blank)
- ">" (Greater than)
- "<=" (Less than or equal to)
- ">=" (Greater than or equal to)
- "NOTBETWEEN" (Not between)
- "ISNOTBLANK" (Is not blank)
- "BETWEEN" (Between)
- "<" (Less than)
- "=" (Equal to)

The text within each set of parentheses shows what appears in the user interface.

Note: When using operators in metadata elements, make sure that the "less than" operator is written as "<" and the "greater than" operator is written as ">".

For example:

```
<perspective name="Offer Contents Expiring within 7 Days ">
<root choiceGroupId="OfferContent">
<simpleViewCriteria attribute="expiringDays" operator="&lt;="
value="7" />
```

- genSysNote: controls whether changing the value of such an attribute should trigger a "Set Attribute" audit trail entry, true or false, true by default
- type (see later in this section)
- restriction (see later in this section)
- default-value: (see later in this section)
- control (see later in this section)

Properties of type:

- **name**: the type of the attribute, either "string", "date", "number", or "clob".
- length: for string types, the maximum number of characters that can be entered into the text control. This includes the characters representing the new line. If set to 0 or less, the maximumLength is ignored. Note that in some browsers like Internet Explorer, new line is treated as two characters.
- precision: for number types, the precision is the total number of digits. It must be between 1 to 15. The default value for precision is 5.
- scale: for number types, the scale is the number of digits to the right of the decimal point. It can range from 0 to the value of precision. The default value for scale is 2.
- displayWidth: the size of the text control specified by the number of characters shown. The number of columns is estimated based on the default font size of the browser.
- required: whether a non-null, non-empty value must be entered.

Properties of restriction:

- **kind**: the kind of type restriction, either "lov" (only for string attributes), "regexp" (only for string attributes), or "range" (only for number and date attributes).
- minDefined: for range type restrictions, whether the range has a min bound ("true" or "false")
- maxDefined: for range type restrictions, whether the range has a max bound ("true" or "false")
- minInclusive: for range type restrictions, whether the range min bound is included ("true" or "false")
- maxInclusive: for range type restrictions, whether the range max bound is included ("true" or "false")

(all other type restriction information is retrieved at runtime from the Inline Service)

Properties of default_value:

The default-value tag has one required attribute "value" and two optional attributes: "expression" and "masterValue".

- **value**: a value of the following types: "string", "date", "number" or a Groovy expression.

  For date attributes, the value must be in the format yyyy-mm-dd, for example, 2012-10-08.

  If a string, date or number value is defined for the "value" attribute, then the optional "expression" attribute must be set to "false" or omitted.

- expression: a Boolean value that indicates whether the "value" attribute specifies a Groovy expression (default is "false"). If the "value" attribute contains a Groovy expression, then expression must be set to "true".

  For example: <default-value value="adf.currentDate" expression="true"..... />

- masterValue: for a project attribute, this specifies the default value for the project attribute in the Main Repository. It may contain a value of the following supported types: string, date, number.

  The following considerations apply for default values for project attributes:

  1. If the "required" attribute (a property of type) is set to "true", then the default-value "value" attribute must be non-empty.
  2. If "expression" is set to "true", then "masterValue" must be non-empty.
  3. During application generation, Oracle RTD Decision Manager sets the required attributes of the Main Repository project based on the default values for these attributes in the project choice group. Expressions cannot be used at that time. So if you use an expression as a default value on a project attribute, you must also add a static value in “masterValue” that will be used for the Main Repository project.

     In other words, for project attributes where the "value" attribute contains a Groovy expression and where "masterValue" is non-empty, then "masterValue" is used for the column value *(for example: <default-value value=”adf.currentDate” expression=”true” masterValue=”2012-08-29” />)*, otherwise the contents of the "value" attribute are used *(for example: <default-value value=”2012-08-30” expression=”false”/>)*.

Properties of control:

- **name**: the widget name, which must be one of the following:
  - Image (see details later in this section)
  - InputDate - this widget creates a text field for entering dates and a glyph which opens a popup for picking dates from a calendar.
  - InputText (see details later in this section)
  - InputTextRange - this widget is used for attributes with Range Type Restriction.
  - InputTextRegexp - this widget is used for attributes with Regexp Type Restriction
  - ManyToManyRelationship - this widget is used to show the shuttle for selecting a choice participating in a many-to-many relationship.
  - OneToManyRelationship - this widget is used to show a dropdown list for selecting a choice participating in a one-to-many relationship.

- OutputRelationship - this widget displays a choice participating in many-to-many or one-to-many relationships.
- OutputText - this widget displays text (read only).
- RTDRuleEditor - (see details later in this section)
- SelectOneChoiceLOV - this widget displays list of values derived from a Type Restriction.

An Image widget is used to display an image when a choice is opened in view mode. The URL location of the image is determined at run time, during choice creation or editing.

An Image widget can have two optional attributes:

- urlPrefix
- urlPostfix

The values of urlPrefix and urlPostfix can be used to form the complete URL where the image is to be found. During choice creation or editing, the location of the image is provided either as a complete URL, or as a value to which the urlPrefix and urlPostfix values, if specified in the metadata, can be attached.

Typically, urlPrefix specifies a general directory that contains multiple images *(for example, http://www.oracleimg.com/us/assets/)*, and choice creators and editors provide the name of a particular image file in that directory *(for example, oralogo-small.gif)*.

An InputText widget is used to enter values for attributes. An InputText widget can have one optional attribute:

- rows: This determines if a text field will have one or more rows. If rows is not specified, the default value is 1

A RTDRuleEditor widget displays the ADF RTD Rule Editor for the creation of rules in Decision Manager.

A RTDRuleEditor widget has the following optional attributes:

- title

  Specifies the message which is shown in the Rule Editor as the top Rule Editor entry. By default, the message is "{0} is eligible". Use {0} to refer to the choice group name.

- object

  Specifies the id of the choice group to which the rule applies. Leave this empty to enable the rule to be for the choice group of the attribute that contains this control.

- inlineService

  Specifies the Inline Service which this rule uses. By default this is the Inline Service specified in config/config.xml.

Properties of relationship types:

- **id**: the id of the relationship type. You use this id in your Inline Service for methods such as CLMChoiceBag.getRelatedChoiceIds().
- **fromChoiceGroupId**: the choice group source.
- **toChoiceGroupId**: the choice group destination.

- **fromName**: the owner to destination relationship type name. You use this name in your Inline Service for methods such as CLMChoiceBag.getRelatedChoiceIds().
- **toName:** the destination to owner relationship type name. You use this name in your Inline Service for methods such as CLMChoiceBag.getRelatedChoiceIds().
- fromDescription: the owner to destination relationship type description.
- toDescription**:** the destination to owner relationship description.
- **cardinality**: From the owner to the destination, can be:
  - ***:1**

    Each source has one destination

    Each destination has 0 to n sources
  - ***:0**

    Each source has 0 or 1 destination

    Each destination has 0 to n sources
  - ***:***

    Each source has 0 to n destinations

    Each destination has 0 to n sources
  - ***:**1..*

    Each source has 1 to n destinations

    Each destination has 0 to n sources

    Note: For cardinalities *:* and *:1..*, the only valid value for onDelete is "O".

- onDelete: 'C' to cascade delete destination choices, 'O' to make destination choices orphans (that is, only delete the relationship). Explicitly:
  - onDelete="C": Deleting a choice on the "zero or one" side of the relationship will attempt to also delete the choices on the "many" side
  - onDelete="O": Deleting a choice on the "zero or one" side of the relationship will not delete the choices on the "many" side

  For cardinalities *:0 and *:1, the default onDelete value is "C".
- propagateRules: 'N' to not propagate rules (the default), 'D' to propagate rules from owner to destination, 'S' to propagate rules from destination to owner, 'B' to propagate rules in both directions. For more details, see the section "Propagation of Rules" in *Oracle Real-Time Decisions Base Application Installation and Reference Guide*.
- propagateEvents: 'N' to not propagate events (the default), 'D' to propagate events from owner to destination, 'S' to propagate events from destination to owner, 'B' to propagate events in both directions. For more details, see the section "Propagation of Events" in *Oracle Real-Time Decisions Base Application Installation and Reference Guide*

Properties of criteria:

- **id**: the id of the criteria

- **xml**: the <ViewCriteria> xml of a view criteria defined on this choice group's VO (view object).

Projects

Projects are a special kind of choice group. You can specify additional attributes to projects by specifying a choice-group with id Project. Project attributes appear in the Decision Manager interface when creating projects, viewing projects and project audit trail entries, and in the model layer, where they can be accessed by the Oracle RTD public Java APIs.

2.2.2.4 User Interface Pages

The layout of the pages and tabs used when creating, editing, and viewing choices and projects is determined by XML metadata. These metadata tags may be in any .xml file in the top-level application implementation folder *(for the reference implementation, this is **clm\Build\metadata\ref**)*, but is typically included in the individual choice group and project XML files.

The metadata tags that control the user interface pages for choices and projects are the following:

- **views**: top-level tag that is the container for the Decision Management user interface layout.
- **choice-view** (within views): the Decision Management user interface view for choice data of a particular choice group.

  Uniquely identified by choiceGroupId and view type. Supported types for choices and projects are "create", "edit", and "view". In addition, the view type "list" is supported for projects.

  Note: When type="list", the choice-view attribute can contain several <table> attributes, each of which can have multiple <column> attributes.

  Each table must have a unique **id** (in the context of the choice-view) and the attribute **viewObjectName** (which defines the VO used for the table). ActiveProjectsVO and DeployedProjectsVO are the supported VOs.

  Each column must have the attribute **attribute-id**. Attributes with type number, date, and string are supported, but not attributes with type clob.

  Columns can have filterable and sortable boolean attributes, for example, <column attribute-id ="Name" filterable ="false" sortable ="false" />. By default, both filterable and sortable are "true" - however, filterable=true is not supported for date attributes.

  For an example, see project.xml for the reference implementation.

- **page** (within choice-view): the page or tab that contains the elements required for an operation to be performed on a choice or project. *The actual contents of the page are specified in the page-attributes-group tag (see later in this section).*

  A page corresponds to either a tab (displayed when choices and projects are edited or viewed) or a train stop (displayed during choice creation).

Each page must have an id and a Name, and may have a description. As well as for identification, the page id is used to assign permissions to a page.

A page can have the viewType attribute, whose value designates which type of view it belongs to and for which operation it is displayed (one or more of the options "create", "edit", and "view"). If viewType is not specified, the page belongs to all the view types and is displayed for create, edit, and view operations.

For example, with <page id="Review" viewType="create edit"/>, the page is displayed on create and edit views only.

- **page-attributes-group** (within page): the attributes, relationships, audit trail notes, and text to be displayed on the page that are required for an operation to be performed on a choice or project. On the page, page attribute groups are separated from each other by a dotted line.

  Every page must have one and only one top level <page-attributes-group/>. Nested occurrences of <page-attributes-group/> are used to group attributes. In the Decision Manager user interface, dotted lines appear where nested page attribute groups are defined in the metadata.

- attribute (within page-attributes-group): corresponds to the choice group or project attribute.

  May contain the readonly and viewType attributes. As with the page tag, viewType controls whether the attribute appears on some or all types of view.

  All attributes for viewType = "view" are readonly.

  Note: The following five built-in attributes are also supported: Name, Description, ChoiceId, EligibilityRule and LockOwner (this last one only when ownership-mode is on in the config XML file.

- relationship (within page-attributes-group): corresponds to the relationship between choice groups.

  May contain the readonly and viewType sub-tags.

  All relationships for viewType = "view" are readonly.

  As with the attribute tag, viewType controls whether the relationship appears on some or all types of view.

- note (within page-attributes-group): renders an Audit Trail Note field.

  Can be required, and, as with the attribute tag, it can have viewType.

  Does not have the readonly option, and is not displayed for viewType = "view".

- text (within page-attributes-group): appears as text message on the page.

2.2.2.4.1 Example of User interface XML for Placement Choice Group

```
<choice-view choiceGroupId="Placement" type="create edit view">
      <page id="Overview">
        <name>Overview</name>
        <description>Overview</description>
        <page-attributes-group>
          <page-attributes-group>
            <attribute id="Name"/>
            <attribute id="Description"/>
            <attribute id="ChoiceId" readonly="true" viewType="edit view"/>
          </page-attributes-group>
```

```
          <page-attributes-group>
            <attribute id="type"/>
            <attribute id="url"/>
          </page-attributes-group>
          <page-attributes-group>
            <relationship id="Placement to Channel"/>
          </page-attributes-group>
          <note viewType="edit"/>
        </page-attributes-group>
      </page>
      <page id="Rules">
        <name>Rules</name>
        <description>Rules</description>
        <page-attributes-group>
          <attribute id="EligibilityRule"/>
          <note viewType="edit"/>
        </page-attributes-group>
      </page>
      <page id="Confirm" viewType="create">
        <name>Confirm</name>
        <description>Confirm</description>
        <page-attributes-group>
          <text value="An Id was automatically generated based on the name you
entered. You can change it now. It cannot be changed later."/>
          <text value="Only letters, digits, underscores, hyphens, dots and spaces
are allowed."/>
          <attribute id="ChoiceId" readonly="false"/>
          <note/>
          <page-attributes-group>
            <text value="Review the information you entered below. Press Back if
you need to make changes, or OK to finish."/>
            <attribute id="Name" readonly="true"/>
            <attribute id="Description" readonly="true"/>
            <attribute id="type" readonly="true"/>
            <attribute id="url" readonly="true"/>
          </page-attributes-group>
          <page-attributes-group>
            <relationship id="Placement to Channel" readonly="true"/>
          </page-attributes-group>
        </page-attributes-group>
      </page>
    </choice-view>
```

2.2.2.5 Security XML Files

By default, all security metadata is in a file called `security.xml` but you may create as many files containing security information as you want.

There are different root nodes that apply to security:

- <application-roles> contains the definition of application roles.
- <enterprise-roles> contains the definition of enterprise roles, which are uploaded as WebLogic groups in a development environment, and can be used in production environment to map application roles to groups you define in the WebLogic or WebSphere console (see Section 1.6.4, "Mapping Users and Groups to Intermediary Roles in Production.")
- <users> contains the definition of users, which are uploaded as WebLogic users in a development environment, but are not meant to be used in a production environment

<application-roles>

<application-roles> contains multiple <application-role>.

Each application role has a <name> and <permissions>.

<permissions> has multiple <permission>, each having a <resource-name>, a <resource-type>, and <actions>.

The resource type can be perspective, choice group, project, choice view page, and cache. Cache is specifically for admin users.

For the perspective resource type:

- The resource name is the name of the perspective as defined in perspectives.xml, or _all_ to mean all perspectives.
- The only action is view, which means that the user having this application role will be able to see and select this perspective in the Decision Manager user interface.

For the choice group resource type:

- The resource name is the id of the choice group as defined in one of the other configuration files, or _all_ to apply to all choice groups.
- The actions are a comma separated list of one or multiples of:
  - create: to be able to create a choice of that choice group in a project
  - read: to be able to view choices of that choice group
  - update: to be able to edit a choice of that choice group in a project, and to be able to discard any changes (including addition and deletion) on a choice of that choice group
  - delete: to be able to delete a choice of that choice group in a project
  - own: to be able to edit (in the same project) and become owner of a choice that is owned by someone else

For the project resource type:

- The resource name must be _all_.
- The actions are a comma separated list of one or multiples of:
  - create: to be able to create a project
  - commit: to be able to commit a project
  - discard: to be able to discard a project and all changes within it
  - update: to be able to update a project
  - read: to be able to log in to the application and to use the Decision Manager user interface

For the choice view page resource type:

- The resource name has the following format {choiceGroupId} / {pageId}.

  Both {choiceGroupId} and {pageId} can be specific Ids or _all_:

  - _all_/_all_ means the permission is granted for all pages in all choice groups
  - _all_/Confirm means the permission is granted for all pages with pageId=Confirm in all choice groups
  - Offer/_all_ means the permission is granted for all pages in the Offer choice group.

- The actions are a comma separated list of one or multiples of:
  - create: to be able to create objects in the pages specified by <resource name>
  - view: to be able to view objects in the pages specified by <resource name>
  - update: to be able to objects in the pages specified by <resource name>
  - list: to be able to view lists of projects in the pages specified by <resource name>

    For "list" the choice group must be Project and the resource name must be a page representing a list of projects to be shown in the View Projects tab.
- Example

  The following <permission> metadata enables all pages of all choice groups to be viewed, and all Overview pages *(technically, pages with pageId=Overview)* in all choice groups to be edited:

```
<permission>
   <resource-type>choice view page</resource-type>
   <resource-name>_all_/_all_</resource-name>
   <actions>view</actions>
</permission>
<permission>
   <resource-type>choice view page</resource-type>
   <resource-name>_all_/Overview</resource-name>
   <actions>update</actions>
</permission>
```

For the cache resource type:

- This permission, to be used only by admin users, is for cleaning the RTD cache. This cache contains the type restrictions information retrieved from the Inline Service.

  Example:

```
<permission>
  <resource-type>cache</resource-type>
  <resource-name>rtd-metadata</resource-name>
  <actions>reset</actions>
</permission>
```

<enterprise-roles>

<enterprise-roles> has multiple <enterprise-role>.

Each <enterprise-role> has a name and <application-roles>, which is a comma separated list of application roles that are granted to members of that enterprise role.

<users>

<users> has multiple <user>.

Each <user> has a name, a display name, a description and credentials which is the password in an encrypted form. See previous sections for how to get the correct credential string.

Each <user> has <application-roles>, which is a comma separated list of application roles that are granted to members of that enterprise role.

2.2.2.6 Folder for Choice Group Images

For each choice group you have defined, add a 16x16 png image in the **images\group** folder (under ***<metadata_modules_home>\<your_application>***) named after the choice group id.

The file must be called `<choiceGroupId>.png`.

Both parts of the file name are case-sensitive:

- <choiceGroupId> must exactly match the name of the choice group id, as set up in the choice group configuration xml file (see Section 2.2.2.3, "Choice Group, Project, and Relationship-Types XML Files")
- png must be all lower-case

2.2.2.7 Inline Service Folder

The RTD_Base_Marketing Inline Service released with the RTD for Marketing Optimization application is in the service folder under ***<metadata_modules_home>*\ref**, that is, in **clm\Build\metadata\ref\service**. This Inline Service is described in more detail in *Oracle Real-Time Decisions Base Application Installation and Reference Guide.*

2.2.3 Ant Tasks

After you have modified the metadata, you need to run the application generation tool ant to generate the application.

The **clm\Build** directory contains the ant tasks to perform the generation.

The ant targets are as follows:

- ear - creates EAR file for deployment in production mode
- clean - cleans projects
- generate - generates the application.
- clean-generated - removes generated code

The ant option "generate" takes two optional parameters:

- metadata.module: this specifies which subfolder of **clm\Build\metadata** must be used to generate the application. The default value is **ref**.
- changes: this specifies whether to warn you and abort the generation if it is about to overwrite files that have been modified. Use value **overwrite** to overwrite files. The default value is to omit this parameter, in which case ant will warn and fail.

Example:

To generate an application in **clm\Build\metadata\myapp**, call:

```
ant -Dmetadata.module=myapp generate
```

Then

`ant ear` (for deploying to production, otherwise deploy from JDeveloper)

Notes:

- After generation, the database schema may have changed. You should then run "drop ils.sql" then "drop core.sql" (or drop and recreate the database user), then "load core.sql" then "load ils.sql". This will delete any choice you have in the database. To preserve choices, you can compare the previous and new sql and just

make these changes. For instance, if you added a choice attribute, you can alter the table to add a column, and recreate the view associated with it.

- In most cases, you should not modify any of the files that are generated by ant generate, so you should not use `-Dchanges=overwrite`. There are a few exceptions, such as the steps described in Section 1.6.1.3, "Running Application Generation."

  Customization can be achieved without modifying any of the files created by ant generate. Use the extensibility of the ADF Framework or the Oracle RTD Decision Management templates to extend the behavior of the application without modifying any of the core and generated files.

2.2.4 Java API

The behavior of the Oracle RTD Decision Management business layer and view controller layer can be further extended by the use of the Java API that comes with it. The javadoc for this Java API can be found in `clm\lib\clm-model-api-javadoc.jar` and also `clm\lib\clm-ui-core-api-javadoc.jar`.

2.2.5 Application Extensions

This section contains the following topics:

- Section 2.2.5.1, "Adding an Attribute to a Choice Group or Project"
- Section 2.2.5.2, "Adding a Choice Group"
- Section 2.2.5.3, "Creating a Relationship Type Between Two Choice Groups"
- Section 2.2.5.4, "Modifying Perspectives"

2.2.5.1 Adding an Attribute to a Choice Group or Project

In order to add an attribute to a choice group or project, follow these steps:

- Modify the xml where the choice group or project is defined in folder **clm\Build\metadata\ref**.
- Run application generation again: in folder **clm\Build**, run "`ant generate`".
- Since adding an attribute to a project or choice group modifies the database, you have to recreate the database. In order to do this, call "drop ils.sql" then "drop core.sql" then "load core.sql" then "load ils.sql" (these files are in **clm\Database\sql** and **clm\Database\sql\ils**). If you do not want to lose your database data, you can compare the sql to the previous one and alter the table to add the column and recreate the associated view.
- You can now deploy the new version of the Decision Management application using JDeveloper.
- For new choice group attributes, you have to add the choice attribute to the choice group in your Inline Service as well. Make sure the id is the same as the one you entered in the xml metadata in the first step. Redeploy the Inline Service.

2.2.5.2 Adding a Choice Group

In order to add a choice group, follow these steps:

- Add an xml file for this new choice group in folder **clm\Build\metadata\ref**.

- Add an image for this choice group in folder **clm\Build\metadata\ref\images\group**.
- Run application generation again: in folder **clm\Build**, run "`ant generate`".
- Since adding a choice group modifies the database, you have to recreate the database. In order to do this, call "drop ils.sql" then "drop core.sql" then "load core.sql" then "load ils.sql" (these files are in **clm\Database\sql** and **clm\Database\sql\ils**). If you do not want to lose your database data, you can compare the sql to the previous one and add the new table, the new view and one row in the CHOICE_GROUP table.
- You can now deploy the new version of the Decision Management application using JDeveloper.
- You have to add the choice group in your Inline Service as well. Make sure the ids are the same as the ones you entered in the xml metadata in the first step. Make sure the choice group you add is under **CLM Base** choice group. Add the choice group in the **CLM ILS Choice Groups** application parameter. Redeploy the Inline Service.

2.2.5.3 Creating a Relationship Type Between Two Choice Groups

In order to create a relationship type between two choice groups, follow these steps:

- Edit `relationship-types.xml` in folder **clm\Build\metadata\ref** and add the new relationship type.
- Run application generation again: in folder **clm\Build**, run "`ant generate`".
- Since adding a choice group modifies the database, you have to recreate the database. In order to do this, call "drop ils.sql" then "drop core.sql" then "load core.sql" then "load ils.sql" (these files are in **clm\Database\sql** and **clm\Database\sql\ils**). If you do not want to lose your database data, you can compare the sql to the previous one and add one row in the RELATIONSHIP_TYPE table.
- You can now deploy the new version of the Decision Management application using JDeveloper.
- You can (but do not have to) add choice attributes in both choice groups in the Inline Service to follow the relationship. For an example, see how Offer has a "campaign" choice attribute and a "creatives" choice attribute. Redeploy the Inline Service.

2.2.5.4 Modifying Perspectives

In order to modify perspectives, follow these steps:

- Edit `perspectives.xml` in folder **clm\Build\metadata\ref\config**.
- Run application generation again: in folder **clm\Build**, run "`ant generate`".
- You can now deploy the new version of the Oracle RTD Decision Management application using JDeveloper.

2.3 Miscellaneous

This section contains the following topics:

- Section 2.3.1, "Configuring Logs"

2.3.1 Configuring Logs

Oracle RTD Decision Management uses Java Logging API. There are two ways to configure Java Logging API (http://java.sun.com/j2se/1.5.0/docs/guide/logging/overview.html):

- Update global logging configuration of the JRE which is used to start WebLogic: *JAVA_HOME*/jre/lib/logging.properties
- Create separate logging properties file (for example c:\src\clm\logging.properties) and pass it in the -Djava.util.logging.config.file argument to the Weblogic.Server startup command.

  To pass this argument append the line:

  ```
  set JAVA_OPTIONS=%JAVA_OPTIONS%
  -Djava.util.logging.config.file=
  c:\src\clm\logging.properties
  ```

  to

  C:\Oracle\Middleware\user_projects\domains*<RTDCLM_domain>*\bin\setDomainEnv.cmd

Sample logging.properties file:

```
# Specify the handlers to create in the root logger
.level= INFO
handlers = weblogic.logging.ServerLoggingHandler
# Register handlers for the oracle.rtd.clm. and its child loggers
oracle.rtd.clm.handlers = java.util.logging.FileHandler,
java.util.logging.ConsoleHandler, weblogic.logging.ServerLoggingHandler
oracle.rtd.clm..useParentHandlers = false
oracle.rtd.clm.level = ALL
#Console handler
java.util.logging.ConsoleHandler.level = INFO
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter
# Set the default logging level for new FileHandler instances
weblogic.logging.ServerLoggingHandler.level = ALL
#File handler
java.util.logging.FileHandler.pattern = %h/java%u.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
java.util.logging.FileHandler.formatter = java.util.logging.SimpleFormatter
java.util.logging.FileHandler.level=ALL
```

This configuration logs message with level INFO to the WebLogic Server Startup Console and all messages to the java*.log file in the user home directory. To change log level for Oracle RTD Decision Management applications replace oracle.rtd.clm.level = ALL

with oracle.rtd.clm.level = SEVERE|WARNING|INFO|CONFIG|FINE|FINER|FINEST.

Additional info can be found at the following site:

http://download.oracle.com/docs/cd/E14571_01/web.1111/e13739/logging_services.htm

For WebSphere, logging can be configured as follows:

1. Login to the WebSphere console.

2. Navigate Troubleshooting > Logs and trace.
3. Open the server where Decision Manager is deployed.
4. Open Change Log Detail Levels.
5. To make a static change to the configuration, click the Configuration tab.

   To change the configuration dynamically, click the Runtime tab.
6. In the All Components tree, right-click oracle.rtd.clm.* (or one of its sub packages), and set the logging level.
7. Click Apply.
8. If you changed the logging level on the Configuration tab then save changes to the configuration and restart the server.

Logs can be found in <*server_log_root*>/<*server_name*>-diagnostic.log (for example /AppServer/profiles/AppSrv01/logs/server1-diagnostic.log).