Oracle® Real-Time Decisions Base Application

Decision Manager Installation and Configuration Guide

Release 3.2

E52398-01

December 2014

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

Oracle Real-Time Decisions Base Application Decision Manager Installation and Configuration Guide, Release 3.2

E52398-01

Primary Author: Oracle Corporation

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

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

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 document describes the new features in Oracle RTD Base Application.

- New Features for Oracle RTD Base Application Release 3.2
- New Features for Oracle RTD Base Application Release 3.1.1

New Features for Oracle RTD Base Application Release 3.2

New features in release 3.2 are:

- Support for Multiple Inline Services
- Support for Slices
- Localization
- New Model Reports
- Decision Analytics Reports Integrated within Decision Manager
- New Performance Goal Editor Widget
- Support for Decision Manager Rule Externalization
- Configurable Tabs

Support for Multiple Inline Services

Multiple inline services can be used at the same time with a single Decision Manager instance. One is used for making changes to the choices and the other ones can be used to get data for reports. This way, you can make changes in a development environment while monitoring the performance of a production system.

Additionally, you no longer need to redeploy the Decision Manager application to make changes, you can now do that within the application.

Support for Slices

You can now define slices of your choice universe and attach specific permissions to these slices. These slices can overlap and you can specify you can view and edit choices in these slices. You could, for instance, say that part of your campaigns, offers and creatives are editable by one group of people but not by another. Meanwhile, a third group may not even see these choices.

Localization

Decision Manager has been localized to the following locales: Japanese, Korean, Portuguese (Brazil), German, Italian, French, Spanish, Chinese (China), Chinese (Taiwan).

New Model Reports

Oracle RTD Decision Manager provides a variety of model reports, both for performance (counts) and model analysis. They are built natively within Decision Manager instead of embedding the Decision Center Reports. As a result, you no longer need to setup SSO between Decision Manager and Decision Center.

The reports available are:

- The *Performance/Counts* report, which covers the functionality of both Performance/Counts and Performance/History of Decision Center model reports.
- The *Analysis/Overview* report, which is a sunburst of a tree map of the model and provides a simple yet powerful view into the model.
- The *Analysis/Best Fit* report, which covers the functionality of the same report in Decision Center.
- The *Analysis/Drivers* report, which covers the functionality of the same report in Decision Center and is easier to read.
- The *Analysis/Quality* report, which covers the functionality of the same report in Decision Center, and a new lift per decile in the table.

Decision Analytics Reports Integrated within Decision Manager

We now provide OBIEE dashboards as part of the Decision Analytics RTD platform feature. Decision Manager ships with dashboards specific to the Base Marketing reference implementation. You can embed the out-of-the box dashboards or any dashboard you build within the Decision Manager user interface.

New Performance Goal Editor Widget

You can now assign weights to performance goals using the performance goal editor widget in Decision Manager.

Support for Decision Manager Rule Externalization

You can now specify full external rules; that is, rules which are stored in database and loaded at runtime. Full use of external filtering rules and external scoring rules have been incorporated. An example of external scoring rules is the Marketing Priority scoring rule defined for offers in the Base Marketing reference application.

Configurable Tabs

You can now specify in metadata the top level tabs shown in the detail section of Decision Manager. You can, for instance, change in which tabs reports are shown or remove the audit trail tab.

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 Manager 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 2.2.6.1, "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 Manager 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 Manager 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 Manager 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 Manager 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 Manager 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 Manager widgets now use this framework. By using this framework, custom widgets will be safeguarded during Oracle RTD Decision Manager upgrades.

Customization Framework

You can use Decision Designer to customize the behavior of Oracle RTD Decision Manager 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 Manager 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 Manager 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 Manager 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 Manager application. With Decision Center reporting enabled, Performance and Analysis tabs appear in the main choice details area of the Oracle RTD Decision Manager 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 Manager and Decision Center.

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

Multiple Oracle RTD Decision Manager Applications Deployable to Same Domain

In this release, you can now deploy many Oracle RTD Decision Manager 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:

- Section 1.1, "Installation Overview"
- Section 1.2, "Installing Oracle RTD Decision Management"
- Section 1.3, "Installing RTD Decision Manager with OBIEE Integration"
- Section 1.4, "Security Configuration"

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 Manager application and then creates an Oracle RTD Decision Manager application EAR file, preparing for the file to be deployed. For more information, see Section 2.2, "Configuring Oracle RTD Decision Management".
- The developer or domain administrator deploys the EAR file to the RTD Decision Manager environment that was setup following either Section 1.2, "Installing Oracle RTD Decision Management" or Section 1.3, "Installing RTD Decision Manager with OBIEE Integration".

1.2 Installing Oracle RTD Decision Management

This section contains the following topics:

- Section 1.2.1, "Preparing for Installation"
- Section 1.2.2, "Creating a WebLogic Domain"
- Section 1.2.3, "Installing Oracle RTD Decision Management"

1.2.1 Preparing for Installation

Oracle RTD Server must be installed prior to installing Oracle RTD Decision Manager. For information on how to install Oracle RTD Server, see the *Oracle Fusion Middleware Administrator's Guide for Oracle Real-Time Decisions*. To prepare to install Oracle RTD Decision Management, use the following steps:

1. Install Oracle Database.
2. Install Oracle WebLogic Server 11gR1. Oracle RTD Decision Manager will be installed in a different middleware home and domain than Oracle RTD Server.

   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. Create a WebLogic domain with a Managed Server, as explained in Section 1.2.2, "Creating a WebLogic Domain."
5. 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).

1.2.2 Creating a WebLogic Domain

For conceptual information about WebLogic domains, see Understanding Oracle WebLogic Server Domains in *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*.

For general information about creating WebLogic domains, see *Oracle Fusion Middleware Creating Domains Using the Configuration Wizard* and in particular, the following sections:

- Starting the Configuration Wizard
- Creating a WebLogic Domain in Graphical Mode
- Configuration Wizard Screens

For a basic WebLogic system, the domain to be created consists of:

- One Administration Server
- One machine
- One cluster (with one Managed Server)

There are several ways to create a WebLogic domain. The following description assumes that you use the graphical mode available through the Configuration Wizard.

You can launch the Configuration Wizard either as described in Starting the Configuration Wizard or by using QuickStart (select **Getting Started with WebLogic Server 10.3.6**).

Then follow the instructions as described in Creating a WebLogic Domain in Graphical Mode.

The following table describes the Configuration Wizard Screens and the options you will use to create a simple, single-cluster WebLogic domain.

Step	Screen	Notes
1	Welcome	Create a new Weblogic Domain.

Step	Screen	Notes
2	Select Domain Source	Select **Generate a domain configured automatically to support the following products** and select the following options. ■ Oracle Enterprise Manager: 11.1.1.0 [oracle_common] ■ Oracle JRF: 11.1.1.0 [oracle_common]
3	Specify Domain Name and Location	Domain name (suggestion): DM_Domain Domain location: (Goes under `MW_HOME\user_projects\domains`)
4	Configure Administrator User Name and Password	Used for all WebLogic administrator processes.
5	Configure Server Start Mode and JDK	Generally Production Mode. For JDK, select either the bundled JDK or the local JDK as previously set up.
6	Select Optional Configuration	Select: ■ Administration Server ■ Managed Servers, Clusters and Machines ■ Deployment and Services
7	Configure the Administration Server	Optionally change the values as required for your environment. Example: Administration Server (AS1) listens on 7001.
8	Configure Managed Servers	Example: Managed Server (MS1) listens on 7003.
9	Configure Clusters	**Configure Clusters** ("C1") for general information about creating cluster.
10	Assign Servers to Clusters	Assign MS1 to C1.
11	Configure Machines	Example: Machine Name MC1.
12	Assign Servers to Machines	Assign MS1 to MC1.
13	Target Deployments to Clusters or Servers window	Select: ■ Cluster as the Target ■ The applications DMS Application#11.1.1.1.0 and wsil-wls ■ Library (which automatically selects all the libraries)
14	Target Services to Clusters or Servers	Specify that all the services should be targeted to the cluster.
15	Configuration Summary and create domain	Review the domain configuration details and create domain.

Your own installation might 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 1.4.2, "Using SSL with Oracle RTD Decision Manager Applications")

1.2.2.1 Scaling Up

Scaling up is the process of adding more managed servers to your system, on the same machine where you set up your basic system. You can do this by adding extra managed servers to the cluster set up in the basic system.

This section describes the process of adding extra managed servers to a cluster.

To create an extra managed server, first clone an existing server, specifying the new Server Listen Address and Server Listen Port.

1. Login to Weblogic Admin console.
2. Select **Domain** then **Environment** then **Servers**. The Summary of Servers page appears.
3. Select existing managed server form the servers table and click on clone button. Clone a Server page would be displayed.
4. Specify the new server name, Server Listen Address and Server Listen Port; click **OK**.
5. Start the new managed server.

When all the new managed server startups have completed successfully, you can log into em and verify Domain (domain name) then Cluster (cluster name) then Managed servers.

1.2.2.2 Scaling Out

To scale out to another (target) machine, perform the following steps:

1. Install WebLogic on the second (target) machine.
2. In `nodemanager.properties`, provide the full hostname for `ListenAddress`.
3. Start Node Manager on the target machine.
4. On the source machine:
   a. Create a Domain Template
   b. Navigate to `weblogic-install-dir`:

      ```
      cd weblogic-install-dir
      ```

   c. Run:

      ```
      weblogic-install-dir/wlserver_10.3/common/bin/pack.sh
      -domain=weblogic-install-dir/user_projects/domains/<dm_domain>
      -managed=true -template=<dm_domain>_template.jar -template_name=<dm_domain>
      ```

      This command creates `<dm_domain>_template.jar` in `weblogic-install-dir/user_projects/domains/`.

   d. Copy that JAR file to the target server in `weblogic-install-dir/`.
5. On the target machine:
   a. Navigate to `weblogic-install-dir/user_projects/`domains and unpack the domain by running this command:

      ```
      weblogic-install-dir/wlserver_10.3/common/bin/unpack.sh
      -domain=weblogic-install-dir/user_projects/domains/<dm_domain>
      ```

```
-template=<dm_domain>_template.ja
```

b. Start the WebLogic Scripting Tool (`wlst.sh` or `wlst.cmd`).

c. Connect to the first (source) machine.

```
connect('weblogic','weblogic','t3://admin-server-host:7001')
```

For example:

```
wls:/offline>
connect('weblogic','weblogic','t3://myHostName.myCompany.com:
7001')
```

d. Enroll the second (target) machine in the existing domain (use `nmEnroll`), as shown here:

```
nmEnroll('weblogic-install-dir/user_projects/domains/DM_domain',
'weblogic-install-dir/wlserver_10.3/common/nodemanager')
```

For example:

```
wls:/oraclertd_domain/serverConfig>
nmEnroll('C:/bea/user_projects/domains/DM_domain',
'C:/bea/wlserver_10.3/common/nodemanager')
```

e. Disconnect from the first machine. by running `disconnect()`.

f. Quit the WebLogic Scripting Tool by running `exit()`.

6. On the source machine:

   a. Clone the existing machine, specifying a new name and new host name or IP address values

   b. Clone the existing server and specify that the new server is on the cloned machine.

7. Start the cloned server.

1.2.3 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 1.2.3.1, "Preparing Your Application for Deployment"
- Section 1.2.3.2, "Oracle RTD Decision Management Database Creation"
- Section 1.2.3.3, "Oracle RTD Decision Management Data Source Setup"
- Section 1.2.3.4, "Storing Credentials to Enable Web Service Calls"
- Section 1.2.3.5, "Oracle RTD Decision Management Application Deployment"
- Section 1.2.3.6, "Inline Service Deployment"

1.2.3.1 Preparing Your Application for Deployment

`RTD_Designer.zip` contains the Base Marketing reference application. If you skip this section, you will install that application.

For instructions on creating your own application and configuring Oracle RTD Decision Manager for your needs, see Chapter 2, "Configuring Oracle RTD Decision Management"

After you have finished configuring Oracle RTD Decision Manager for your needs, you will have generated the proper database schema files and the EAR file that will be deployed to the application server.

1.2.3.2 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: Make sure your database uses the AL32UTF8 character set.

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

   - Go to the subfolder `clm\Database\sql`, then run the file `load core.sql`.

   Note: The file `load core.sql` contains generic commands, required for all Oracle RTD Decision Manager applications. The commands in `load ils.sql`, are specific to the application. By default they will contain the commands for the RTD for Marketing Optimization application; however, they would be different if you configured another application.

   - Go to the subfolder `clm\Database\sql\ils` and run the file `load ils.sql`.

     Make sure you have committed your changes.

1.2.3.3 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.
2. Enter the administrator username and password, which was specified during domain creation (see Section 1.2.2, "Creating a WebLogic Domain").
3. Open the Admin Server Console.
4. Log in with the administrator username and password.
5. Click **Lock & Edit** to make change.
6. Navigate the path Services then Data Sources then New then 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**

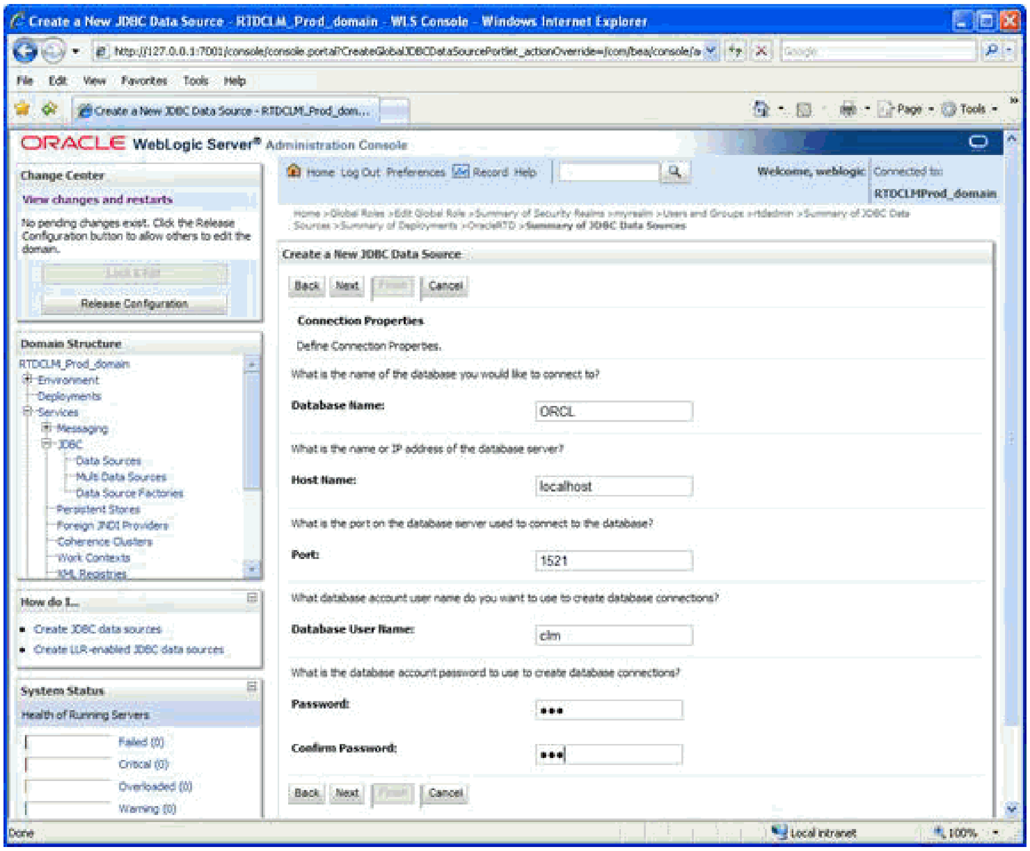
Click Next.

7. For the database driver, use either **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later** or **Oracle's Driver (Thin) for Service connections; Versions: 9.0.1 and later**. Note that, however, for 12c databases you can only use the latter. 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:



Create a New JDBC Data Source

Connection Properties

Define Connection Properties.

What is the name of the database you would like to connect to?

Database Name: ORCL

What is the name or IP address of the database server?

Host Name: localhost

What is the port on the database server used to connect to the database?

Port: 1521

What database account user name do you want to use to create database connections?

Database User Name: clm

What is the database account password to use to create database connections?

Password: •••

Confirm Password: •••

Create a New JDBC Data Source - RTDCLM_Prod_domain - WLS Console - Windows Internet Explorer
File Edit View Favorites Tools Help
ORACLE WebLogic Server® Administration Console
Home Log Out Preferences Record Help
Welcome, weblogic
Connected to: RTDCLMProd_domain
Change Center
View changes and restarts
No pending changes exist. Click the Release Configuration button to allow others to edit the domain.
Lock & Edit
Release Configuration
Domain Structure
RTDCLM_Prod_domain
Environment
Deployments
Services
Messaging
JDBC
Data Sources
Multi Data Sources
Data Source Factories
Persistent Stores
Foreign JNDI Providers
Coherence Clusters
Work Contexts
XML Registries
How do I...
Create JDBC data sources
Create LLR-enabled JDBC data sources
System Status
Health of Running Servers
Failed (0)
Critical (0)
Overloaded (0)
Warning (0)
Back
Next
Finish
Cancel
Done
Local intranet
100%

   Ensure that the Database User name and Password match the values that you set up in step 1 of Section 1.2.3.2, "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 1.2.2, "Creating a WebLogic Domain" as the target, then click **Finish**.

12. Open the data source and select the Connection Pool tab.
13. Go to Advanced Section and deselect **Wrap Data Types** and save.
14. Select **Activate Changes** and then restart servers, if necessary.

1.2.3.4 Storing Credentials to Enable Web Service Calls

Oracle RTD Decision Manager makes web service calls from the Decision Manager server side to Oracle RTD.

On the Oracle RTD side (named RTD_Domain below), we define a user and make him part of an Oracle RTD group, granting him access to Oracle RTD.

On the Decision Manager side (named DM_Domain below), we store the username and password for that user in the WebLogic credential store. It will be used to make calls to Oracle RTD.

Create the User in the RTD_Domain

To create the user in the RTD_Domain, do the following

1. Open the Admin Server Console in the RTD_Domain.
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 create a new user by clicking **New**.
5. Specify a username and password.
6. Click **OK**.
7. Select the user that you just created and then select the Groups tab.
8. Select the RTDDCUserGroup group and shuttle it to the right.
9. Click **Save**.

Store User Credentials In the DM_Domain Weblogic Credential Store

To store the user credentials in the WebLogic credential store in the DM_Domain, do the following:

1. Open the Enterprise Manager.
2. Log in with the administrator username and password.
3. In the Target Navigation Pane, select WebLogic Domain then DM_Domain.
4. In the DM_Domain window, open the DM_Domain dropdown menu and select Security then Credentials.
5. In the Credentials area, click **Create Map**.
6. In the Create Map window, enter `oracle.rtd.clm` and click **OK**.
7. Click **Create Key** then, in the Create Key window:
   - Enter the Key `clm.client` for the map that you just created.
   - Enter the same user name and password you created in the RTD_Domain.
   - Click **OK**

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

1.2.3.5 Oracle RTD Decision Management Application Deployment

The final step is to deploy the Oracle RTD Decision Manager application. This is done by deploying the `clm.ear` file in the `clm/deploy` folder.

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

1. Start 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 then DM_Domain.
5. In the DM_Domain window, from the WebLogic Domain dropdown menu, select Control then **Start Up**.
6. In the DM_Domain window, from the WebLogic Domain dropdown menu, select Application Deployment then 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`.

1.2.3.6 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.

Using Decision Studio, deploy that Inline Service to the Oracle RTD instance running on the same server as Oracle RTD Decision Management.

1.3 Installing RTD Decision Manager with OBIEE Integration

This chapter contains the following topics.

- Section 1.3.1, "Prerequisites"
- Section 1.3.2, "Configuring OBIEE for Decision Manager"
- Section 1.3.3, "BI Analytics Security Configuration"

- Section 1.3.4, "BI Dashboard Metadata"
- Section 1.3.5, "Decision Manager Configuration Metadata"
- Section 1.3.6, "Configure the BI Connection Credentials"
- Section 1.3.7, "Deploy Decision Manager"

1.3.1 Prerequisites

Before you can install Decision Manager with OBIEE, you must met the following prerequisites:

1. Install Oracle Database.
2. Install Oracle JDeveloper 11*g*.
3. Install and configure Oracle RTD. For instructions on how to install Oracle RTD Server, see the *Oracle® Real-Time Decisions Installation and Administration Guide*.
4. Install and configure Oracle BIEE 11*g*. Decision Manager will be deployed in the OBIEE domain and can be integrated with an OBIEE single instance or clustered environment. Configuration in Decision Manager for OBIEE integration is common for OBIEE single instance/OBIEE cluster.

1.3.2 Configuring OBIEE for Decision Manager

The OBIEE integration setup for Decision Manager consists of RPD and Web Catalog.

RPD that contains physical/model/presentation layer for the report schema. This RPD also contains marketing specific measures such as acceptance rate and conversion rate. The RPD contains 2 entries in the physical layer, one containing the model snapshot physical schema and one containing the decision analytics report schema. Two variables are used to define the connection to the report schema: RTD_REPORT_DSN and RTD_REPORT_USERNAME. The password for the RPD is Admin123.

For instructions on configuring OBIEE for Decision Manager, see Section 18.3, "Installation" in *Oracle Real-Time Decisions Platform Developer's Guide*.

1.3.3 BI Analytics Security Configuration

Security configuration needs to be done in BI analytics to ensure that a logged-in user in Decision Manager can access BI analytics through impersonation which would secure an application that uses an Oracle BI EE Presentation Services connection and includes Oracle BI EE objects.

Create a user in the OBIEE domain that will be used for impersonation, as described in Section 1.9.1 "How to Create and Use Impersonate User" in the *Oracle Fusion Middleware Developer's Guide for Oracle Business Intelligence Enterprise Edition*.

1.3.4 BI Dashboard Metadata

The Decision Manager metadata has been enhanced to allow specifying integrating a BI dashboard within a tab. The syntax is:

```
<bi-dashboard path="<path>">
     <parameter name="<name>" value="<value>"/>
              …
</bi-dashboard>
```

Where *<path>* is the path to the dashboard in the web catalog; for example, `path="/<shared/Decision Analytics - Base Marketing/Creatives/_portal/Creative Performance Dashboard".`

Decision Manager supports multiple parameter name/value pairs.

Name	Value (example)	Description
`Applications."App Name"`	`#{biParameters.serviceName}`	The inline service name, app_name_id in the reporting schema
`Choices.CHOICE_ID`	`#{biParameters.choiceId}` (when looking at a creative choice)	The creative id, choice_id in the reporting schema
`Choices.HIERARCHY_ID_1`	`#{biParameters.choiceId}` (when looking at a campaign choice)	The campaign id, hierarchy_id_1 in the reporting schema
`Choices.HIERARCHY_ID_2`	`#{biParameters.choiceId}` (when looking at a offer choice)	The offer id, hierarchy_id_2 in the reporting schema
`Choices.HIERARCHY_ID_3`	`#{biParameters.choiceId}` (when looking at a channel choice)	The channel id, hierarchy_id_3 in the reporting schema
`Choices.HIERARCHY_ID_4`	`#{biParameters.choiceId}` (when looking at a placement choice)	The placement id, hierarchy_id_4 in the reporting schema
`Choices.HIERARCHY_ID_5`	`#{biParameters.choiceId}` (when looking at a slot choice)	The slot id, hierarchy_id_5 in the reporting schema
`Choices.HIERARCHY_ID_6`	`#{biParameters.choiceId}` (when looking at a slot type choice)	The slot type id, hierarchy_id_6 in the reporting schema
dashboard.variables`["DMFrequency"]`	`#{biParameters.timeInterval}`	The time interval of the report
Time."Calendar Date”	`#{biParameters.localizedStartDateString},#{biParameters.localizedEndDateString}"`	The start and end dates of the report

1.3.5 Decision Manager Configuration Metadata

The following metadata has been added to the metadata module’s `config/config.xml`:

```
<bi-server url="http://<server>:<port>/analytics" wsdl-context="analytics-ws" />
```

To use SSL between Decision Manager and the OBIEE server, specify the URL as `"https://`*`<server>`*`:`*`<secure port>`*`/analytics"`.

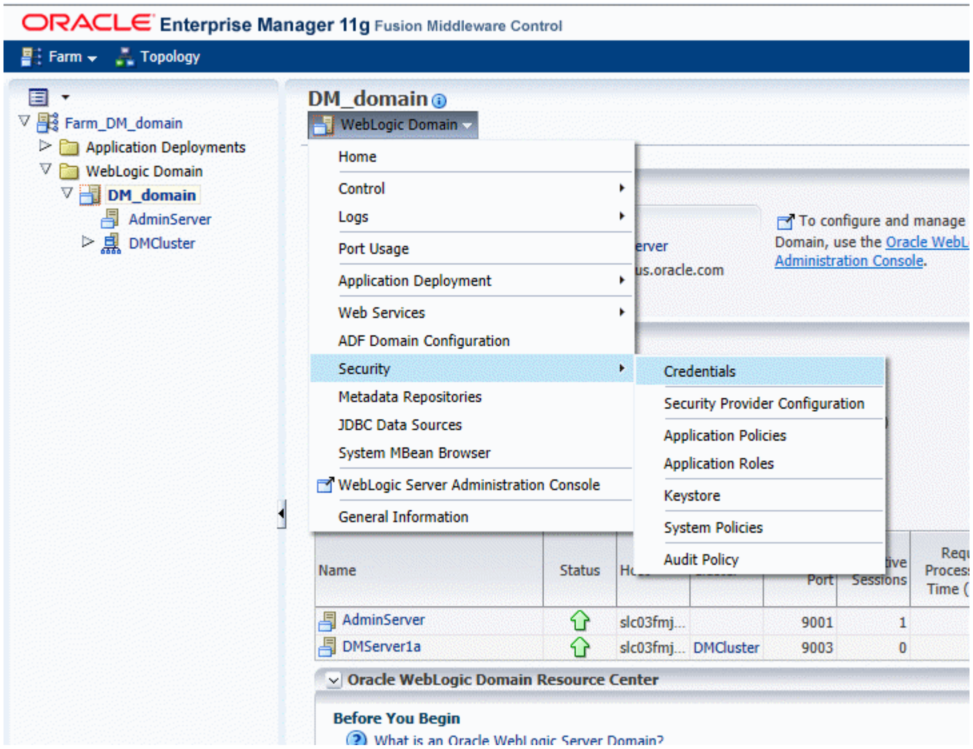
This metadata is optional. For more information, see Section 2.2.2.1, "Config XML File".

If you do not have a `<bi-server>` element, any page that contains a `<bi-dashboard>` element will be hidden.

1.3.6 Configure the BI Connection Credentials

Configure the BI connection credentials before deploying the Decision Manager application:

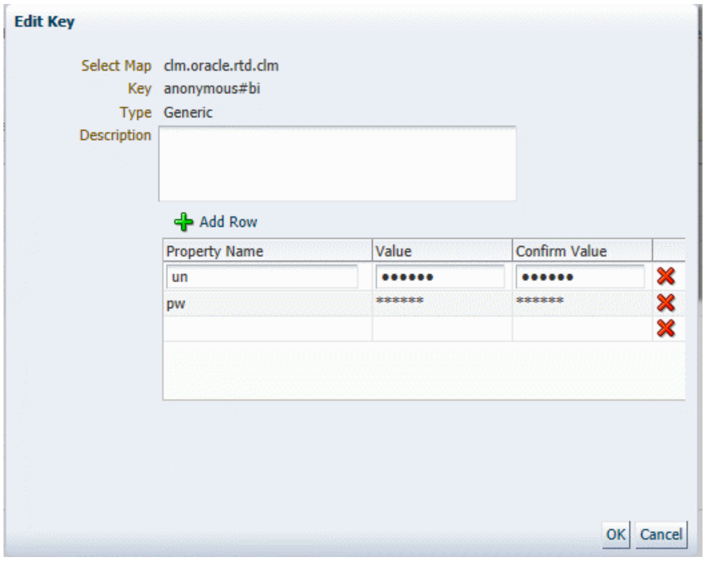
1. Open Enterprise Manager in the Decision Manager domain, select Credentials for the domain, as shown in the following illustration:



2. Create a map called `clm.oracle.rtd.clm`.

3. In that map, create a key called `anonymous#bi`, specify its type to be Generic, and add two rows for the following two properties corresponding to the impersonated user created in Section 1.3.3, "BI Analytics Security Configuration":

   - `un`: username
   - `pw`: password



Edit Key

Select Map clm.oracle.rtd.clm
Key anonymous#bi
Type Generic
Description

Add Row

Property Name	Value	Confirm Value
un	••••••	••••••
pw	******	******

OK Cancel

1.3.7 Deploy Decision Manager

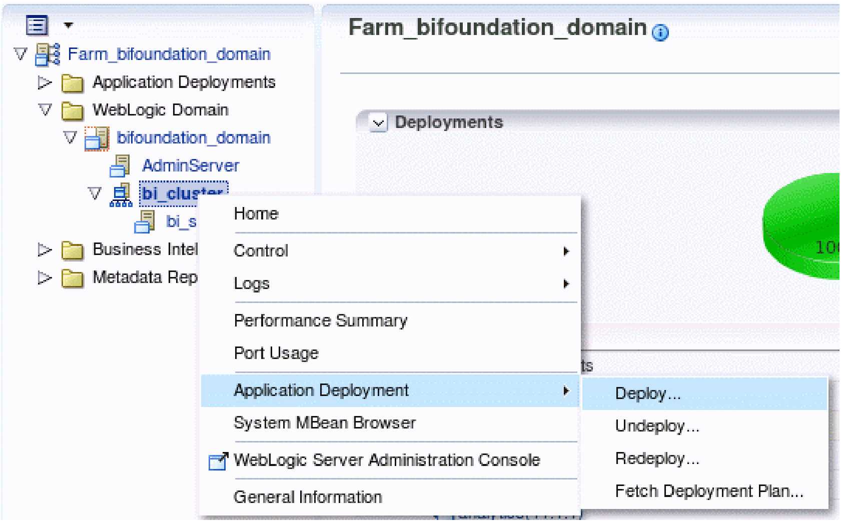
This section explains how to develop, configure and deploy Decision Manager with a BI dashboard by using Enterprise Manager EM.

Before deploying Decision Manager, be aware that you will need to do the following the following:

- Database setup
- Data source creation
- Storing credentials for a web service call
- Inline Service deployment

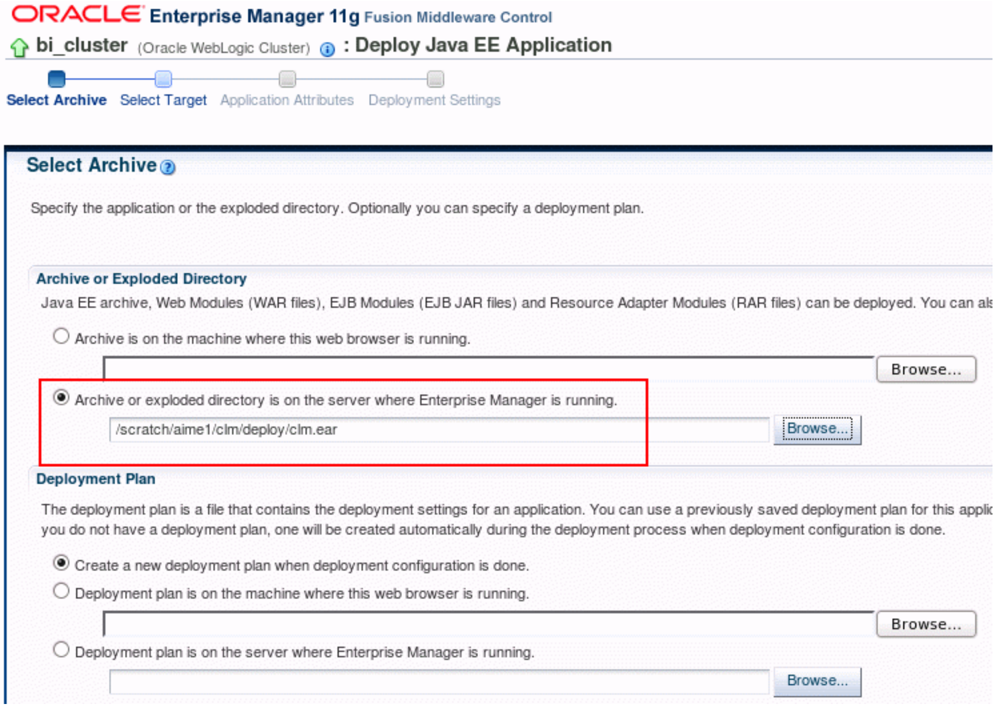
For instructions, see the Section 1.2.3, "Installing Oracle RTD Decision Management".

After that, log in to the Enterprise Manager console and select **Domain** then **Cluster/Server**.



Now, deploy Decision Manager by doing the following:

1. Select the `clm.ear` location, as shown in the following example, and click **Next**.



ORACLE Enterprise Manager 11g Fusion Middleware Control

bi_cluster (Oracle WebLogic Cluster) : Deploy Java EE Application

Select Archive — Select Target — Application Attributes — Deployment Settings

Select Archive

Specify the application or the exploded directory. Optionally you can specify a deployment plan.

Archive or Exploded Directory

Java EE archive, Web Modules (WAR files), EJB Modules (EJB JAR files) and Resource Adapter Modules (RAR files) can be deployed. You can als

○ Archive is on the machine where this web browser is running.

[ ] Browse...

● Archive or exploded directory is on the server where Enterprise Manager is running.

/scratch/aime1/clm/deploy/clm.ear Browse...

Deployment Plan

The deployment plan is a file that contains the deployment settings for an application. You can use a previously saved deployment plan for this applic you do not have a deployment plan, one will be created automatically during the deployment process when deployment configuration is done.

● Create a new deployment plan when deployment configuration is done.

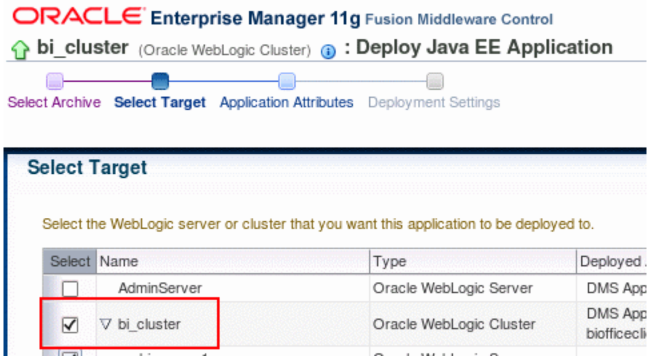
○ Deployment plan is on the machine where this web browser is running.

[ ] Browse...

○ Deployment plan is on the server where Enterprise Manager is running.

[ ] Browse...

2. Select the cluster to which the EAR needs to be deployed and click **Next**.



ORACLE Enterprise Manager 11g Fusion Middleware Control

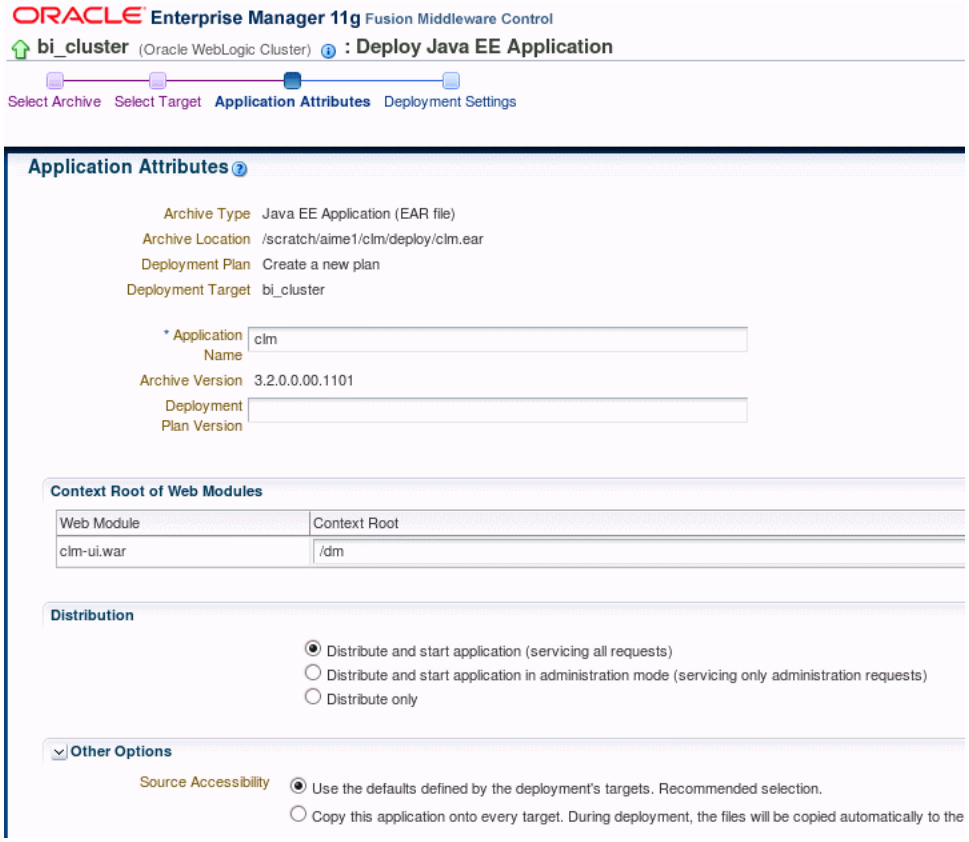
bi_cluster (Oracle WebLogic Cluster) : Deploy Java EE Application

Select Target

Select the WebLogic server or cluster that you want this application to be deployed to.

Select	Name	Type	Deployed
☐	AdminServer	Oracle WebLogic Server	DMS App
☑	▽ bi_cluster	Oracle WebLogic Cluster	DMS App biofficecli

3. Use the default selection or make the necessary changes and click **Next**.



ORACLE Enterprise Manager 11g Fusion Middleware Control

bi_cluster (Oracle WebLogic Cluster) : Deploy Java EE Application

Application Attributes

Archive Type Java EE Application (EAR file)

Archive Location /scratch/aime1/clm/deploy/clm.ear

Deployment Plan Create a new plan

Deployment Target bi_cluster

* Application Name clm

Archive Version 3.2.0.0.00.1101

Deployment Plan Version

Context Root of Web Modules

Web Module	Context Root
clm-ui.war	/dm

Distribution

● Distribute and start application (servicing all requests)

○ Distribute and start application in administration mode (servicing only administration requests)

○ Distribute only

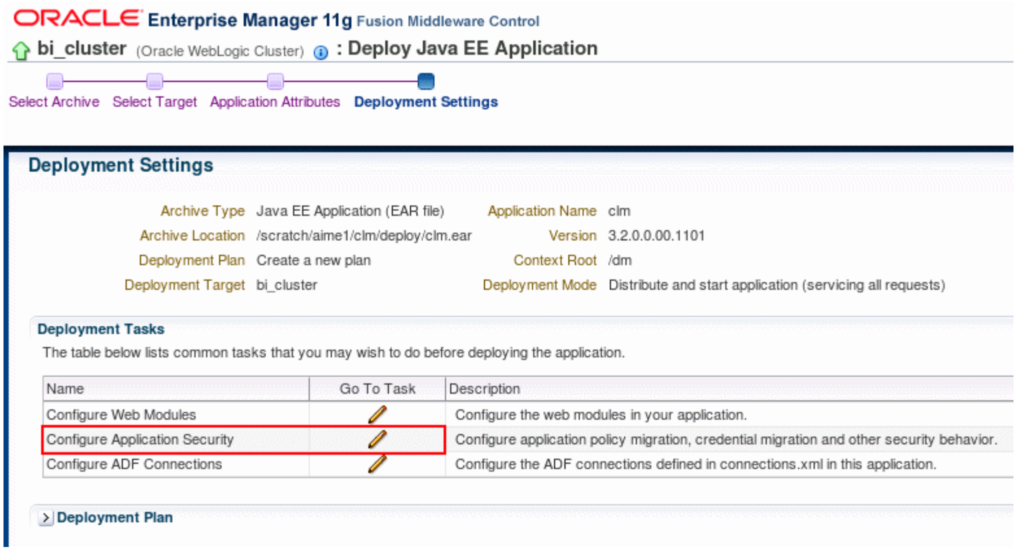
Other Options

Source Accessibility

● Use the defaults defined by the deployment's targets. Recommended selection.

○ Copy this application onto every target. During deployment, the files will be copied automatically to the

4. On the Deployment Settings screen, edit Configure Application Security by clicking ✎ in the Go To Task column.



ORACLE Enterprise Manager 11g Fusion Middleware Control
bi_cluster (Oracle WebLogic Cluster) : Deploy Java EE Application
Select Archive
Select Target
Application Attributes
Deployment Settings

Deployment Settings

Archive Type	Java EE Application (EAR file)	Application Name	clm
Archive Location	/scratch/aime1/clm/deploy/clm.ear	Version	3.2.0.0.00.1101
Deployment Plan	Create a new plan	Context Root	/dm
Deployment Target	bi_cluster	Deployment Mode	Distribute and start application (servicing all requests)

Deployment Tasks

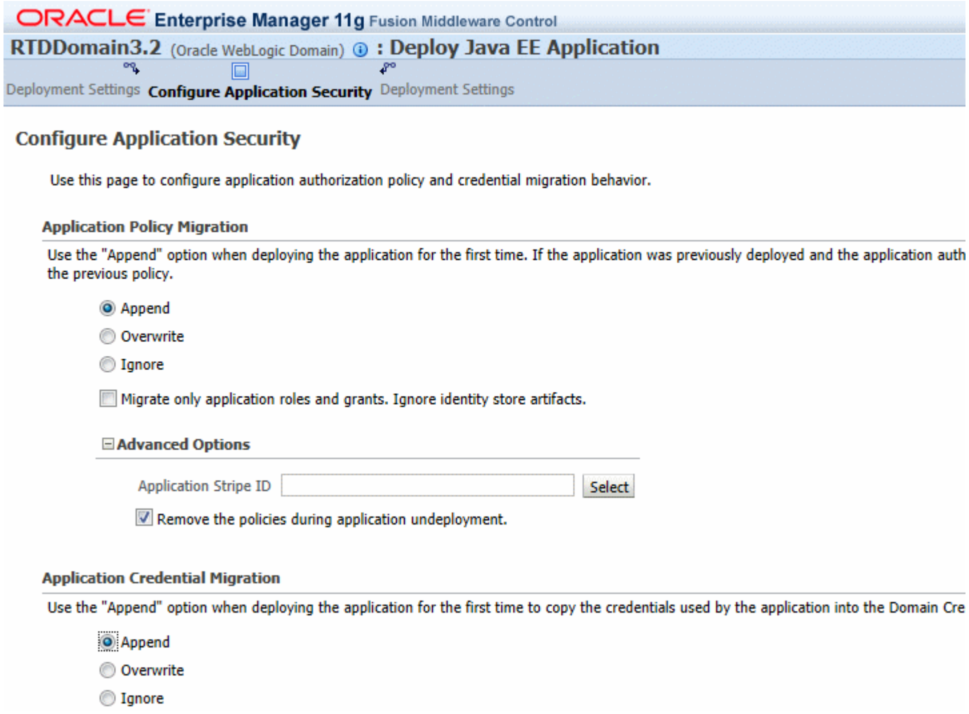
The table below lists common tasks that you may wish to do before deploying the application.

Name	Go To Task	Description
Configure Web Modules	✎	Configure the web modules in your application.
Configure Application Security	✎	Configure application policy migration, credential migration and other security behavior.
Configure ADF Connections	✎	Configure the ADF connections defined in connections.xml in this application.

5. For Migration, in Application Policy, select **Append** (rather than **Overwrite**) and click **Apply**. Use the Append options for first-time deployment only; for subsequent deployments, select **Overwrite**.

6. Depending on the specific circumstances, do the following:

   - If `cwallet.sso` is deployed through `clm.ear` then following page would be displayed:



ORACLE Enterprise Manager 11g Fusion Middleware Control
RTDDomain3.2 (Oracle WebLogic Domain) : Deploy Java EE Application
Deployment Settings
Configure Application Security
Deployment Settings

Configure Application Security

Use this page to configure application authorization policy and credential migration behavior.

Application Policy Migration

Use the "Append" option when deploying the application for the first time. If the application was previously deployed and the application auth the previous policy.

● Append
○ Overwrite
○ Ignore

□ Migrate only application roles and grants. Ignore identity store artifacts.

Advanced Options

Application Stripe ID [ ] Select

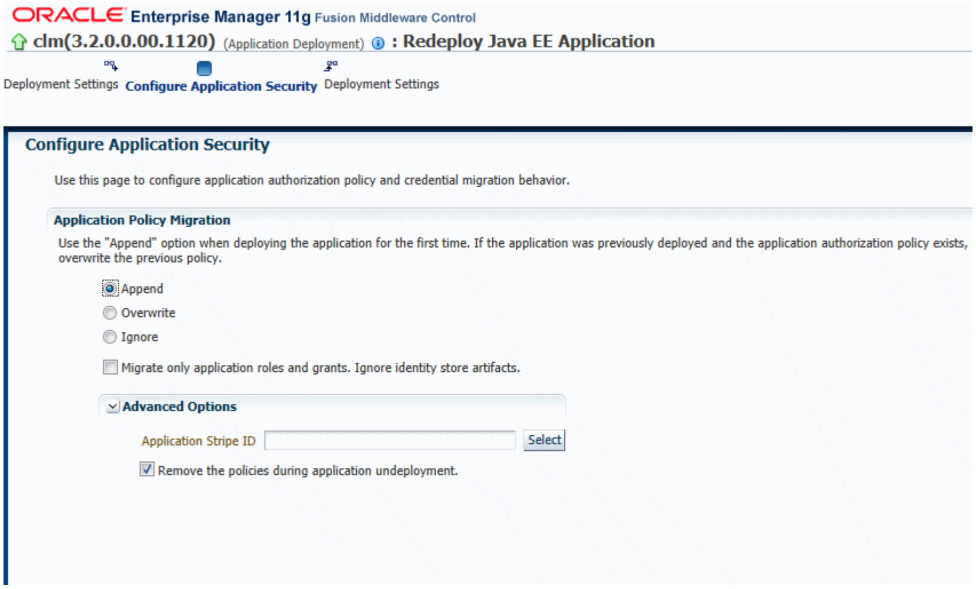
☑ Remove the policies during application undeployment.

Application Credential Migration

Use the "Append" option when deploying the application for the first time to copy the credentials used by the application into the Domain Cre

● Append
○ Overwrite
○ Ignore

   - If credentials are not configured and `cwallet.sso` is not deployed through `clm.ear` then following page would appear; note that Application Credential Migration option does not occur. Select **Append** and click **Apply**.



ORACLE Enterprise Manager 11g Fusion Middleware Control

clm(3.2.0.0.00.1120) (Application Deployment) : Redeploy Java EE Application

Configure Application Security

Use this page to configure application authorization policy and credential migration behavior.

Application Policy Migration

Use the "Append" option when deploying the application for the first time. If the application was previously deployed and the application authorization policy exists, overwrite the previous policy.

● Append

○ Overwrite

○ Ignore

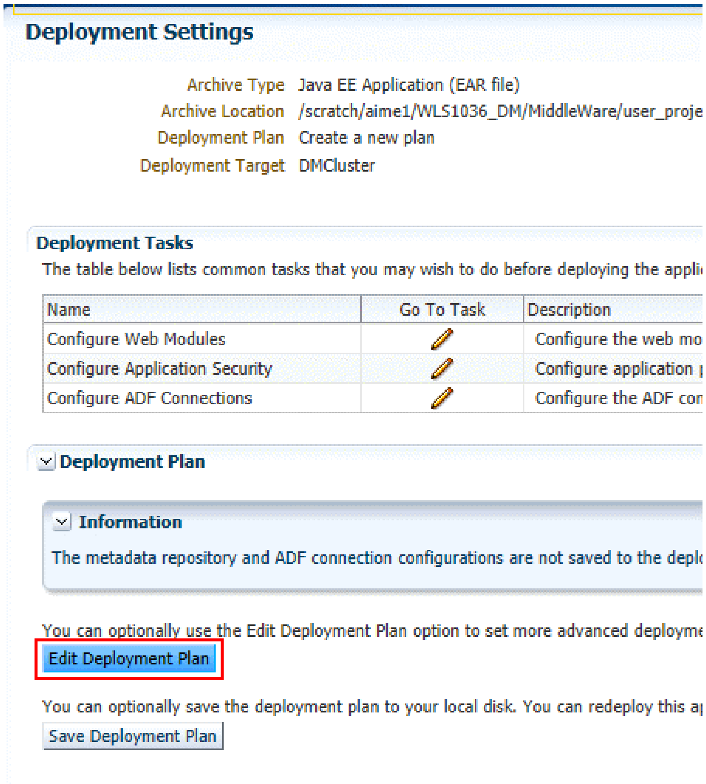
□ Migrate only application roles and grants. Ignore identity store artifacts.

Advanced Options

Application Stripe ID Select

☑ Remove the policies during application undeployment.

- If the credentials option is not available in Configure Application Security, click **Edit Deployment Plan**.



Deployment Settings

Archive Type	Java EE Application (EAR file)
Archive Location	/scratch/aime1/WLS1036_DM/MiddleWare/user_proje
Deployment Plan	Create a new plan
Deployment Target	DMCluster

Deployment Tasks

The table below lists common tasks that you may wish to do before deploying the appli

Name	Go To Task	Description
Configure Web Modules	✎	Configure the web mo
Configure Application Security	✎	Configure application
Configure ADF Connections	✎	Configure the ADF con

Deployment Plan

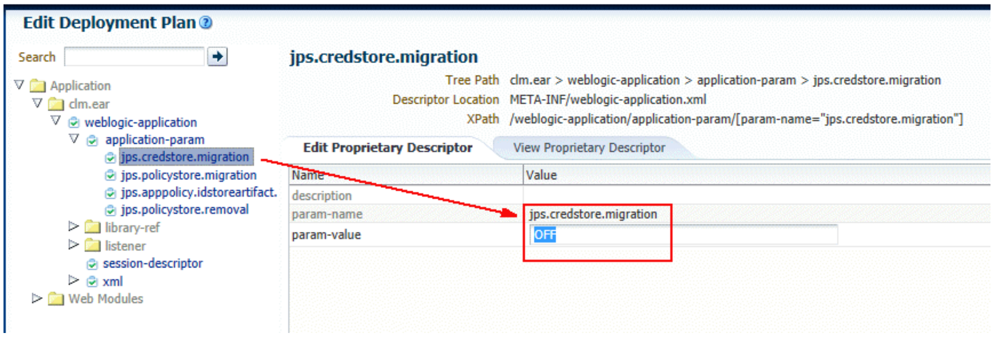
Information

The metadata repository and ADF connection configurations are not saved to the depl

You can optionally use the Edit Deployment Plan option to set more advanced deployme

You can optionally save the deployment plan to your local disk. You can redeploy this a

- Set Credential Store migration parameter value to **OFF** as shown in the following illustration and click **Apply**.

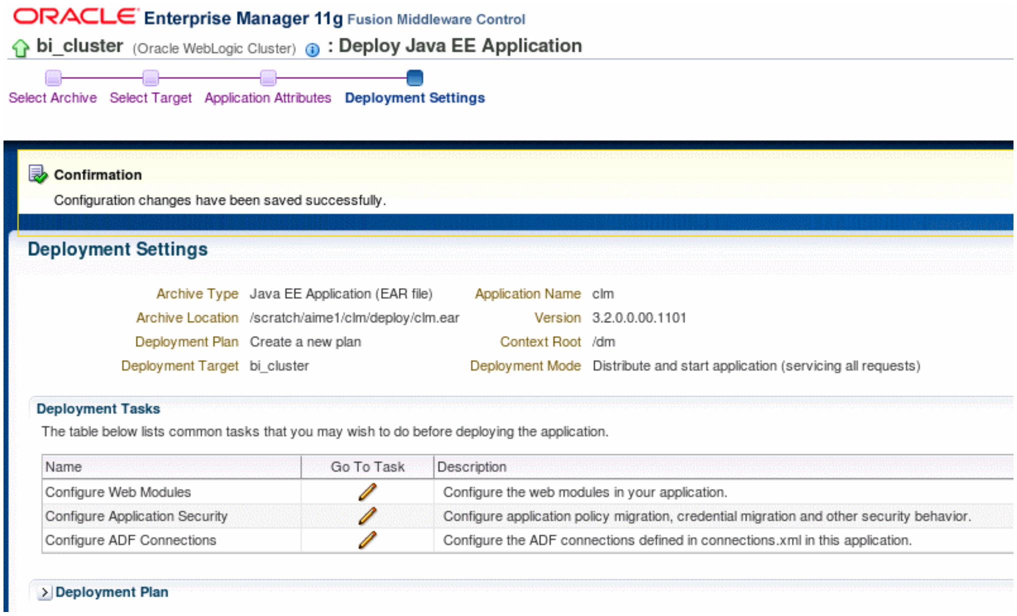


jps.credstore.migration

Tree Path clm.ear > weblogic-application > application-param > jps.credstore.migration
Descriptor Location META-INF/weblogic-application.xml
XPath /weblogic-application/application-param/[param-name="jps.credstore.migration"]

Name	Value
description	
param-name	jps.credstore.migration
param-value	OFF

This saves the settings and the following screen appears:


ORACLE Enterprise Manager 11g Fusion Middleware Control
bi_cluster (Oracle WebLogic Cluster) : Deploy Java EE Application
Select Archive
Select Target
Application Attributes
Deployment Settings
Deployment Plan

Confirmation

Configuration changes have been saved successfully.

Deployment Settings

Archive Type Java EE Application (EAR file)
Archive Location /scratch/aime1/clm/deploy/clm.ear
Deployment Plan Create a new plan
Deployment Target bi_cluster
Application Name clm
Version 3.2.0.0.00.1101
Context Root /dm
Deployment Mode Distribute and start application (servicing all requests)

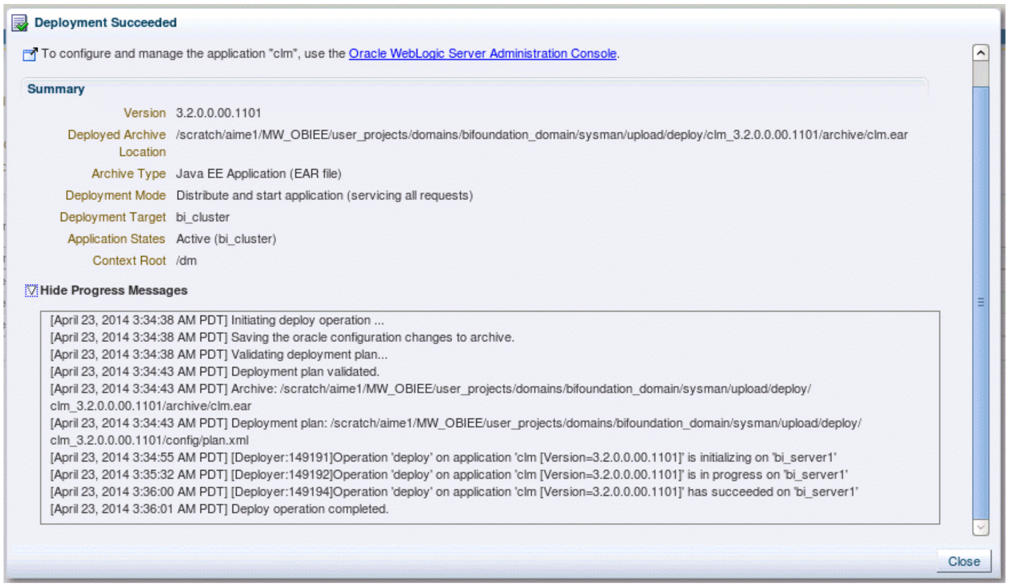
Deployment Tasks

The table below lists common tasks that you may wish to do before deploying the application.

Name	Go To Task	Description
Configure Web Modules	✎	Configure the web modules in your application.
Configure Application Security	✎	Configure application policy migration, credential migration and other security behavior.
Configure ADF Connections	✎	Configure the ADF connections defined in connections.xml in this application.

7. Click **Deploy**.

   `clm.ear` deployment starts. If it succeeds, a screen similar to this example appears:



Deployment Succeeded

To configure and manage the application "clm", use the Oracle WebLogic Server Administration Console.

Summary

Version 3.2.0.0.00.1101
Deployed Archive Location /scratch/aime1/MW_OBIEE/user_projects/domains/bifoundation_domain/sysman/upload/deploy/clm_3.2.0.0.00.1101/archive/clm.ear
Archive Type Java EE Application (EAR file)
Deployment Mode Distribute and start application (servicing all requests)
Deployment Target bi_cluster
Application States Active (bi_cluster)
Context Root /dm

Hide Progress Messages

```
[April 23, 2014 3:34:38 AM PDT] Initiating deploy operation ...
[April 23, 2014 3:34:38 AM PDT] Saving the oracle configuration changes to archive.
[April 23, 2014 3:34:38 AM PDT] Validating deployment plan...
[April 23, 2014 3:34:43 AM PDT] Deployment plan validated.
[April 23, 2014 3:34:43 AM PDT] Archive: /scratch/aime1/MW_OBIEE/user_projects/domains/bifoundation_domain/sysman/upload/deploy/
clm_3.2.0.0.00.1101/archive/clm.ear
[April 23, 2014 3:34:43 AM PDT] Deployment plan: /scratch/aime1/MW_OBIEE/user_projects/domains/bifoundation_domain/sysman/upload/deploy/
clm_3.2.0.0.00.1101/config/plan.xml
[April 23, 2014 3:34:55 AM PDT] [Deployer:149191]Operation 'deploy' on application 'clm [Version=3.2.0.0.00.1101]' is initializing on 'bi_server1'
[April 23, 2014 3:35:32 AM PDT] [Deployer:149192]Operation 'deploy' on application 'clm [Version=3.2.0.0.00.1101]' is in progress on 'bi_server1'
[April 23, 2014 3:36:00 AM PDT] [Deployer:149194]Operation 'deploy' on application 'clm [Version=3.2.0.0.00.1101]' has succeeded on 'bi_server1'
[April 23, 2014 3:36:01 AM PDT] Deploy operation completed.
```

8. Using your Decision Manager log-in credentials, access the Decision Manager application.

   Oracle RTD Decision Manager is now deployed and accessible at this URL:

   ```
   http://<server>:<port>/dm.
   ```

1.4 Security Configuration

This section describes how to map your enterprise users and groups to Decision Manager application roles using Enterprise Manager. It also describes how to setup SSL to secure connections between the end user browser and the application server. Setting up SSL is recommended but not required.

This section contains the following topics:

- Section 1.4.1, "Mapping Users or Groups to Application Roles"
- Section 1.4.2, "Using SSL with Oracle RTD Decision Manager Applications"

1.4.1 Mapping Users or Groups to Application Roles

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.

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 your Decision Manager deployment: Application Deployments then Internal Applications then the name of your Decision Manager application.
4. In the application window, from the Application Deployment dropdown menu, select Security then Application Roles and click **Search application roles**.
5. Click CLMAdministrator, CLMAuthor, CLMConsumer or the customer roles you have created, depending on what permissions you want to give to this user.
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 A**dd roles** and change Application role to Group in the Type dropdown list.
11. Search for your group and click **OK**.
12. Click **OK**.

1.4.2 Using SSL with Oracle RTD Decision Manager Applications

Depending on your configuration, do the following:

- Section 1.4.2.1, "Using SSL without BI Dashboards"
- Section 1.4.2.2, "Using SSL with BI Dashboards"

1.4.2.1 Using SSL without BI Dashboards

To use SSL with your Oracle RTD Decision Manager applications, do the following:

- Configure Oracle RTD Server to use SSL (see Section 2.5, "Using SSL with Oracle Real-Time Decisions" and Section 3.5, "Configuring SSL for Real-Time Decision Server" of the *Oracle Fusion Middleware Administrator's Guide for Oracle Real-Time Decisions*).
- Configure SSL on the Decision Manager Weblogic domain, as described in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.
- Make sure the Decision Manager JVM trust store trusts the certificate authority that Oracle RTD Server uses.

> **Note:** Use a real certificate and not the one in the RTD demo trust store. Configure Decision Studio, Load Generator and other RTD clients to use the right trust store by specifying the `javax.net.ssl.trustStore` as a Java command-line argument.

- Use HTTPS instead of HTTP in the Inline Service configuration (either in the configuration file, see Section 2.2.2.1, "Config XML File", or in the Inline Service Configuration Dialog, see Section A.12.1, "Inline Service Configuration" in the *Oracle Real-Time Decisions Base Application Decision Management Applications User's Guide*).

1.4.2.2 Using SSL with BI Dashboards

To use SSL with your Oracle RTD Decision Manager applications, do the following:

- Configure Oracle RTD Server to use SSL (see Section 2.5, "Using SSL with Oracle Real-Time Decisions" and Section 3.5, "Configuring SSL for Real-Time Decision Server" of *Oracle Fusion Middleware Administrator's Guide for Oracle Real-Time Decisions*).
- Configure Oracle BIEE Server to use SSL (see Chapter 5, "SSL Configuration in Oracle Business Intelligence" in *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*.
- Configure SSL on the Decision Manager Weblogic domain, see the *Oracle Fusion Middleware Securing Oracle WebLogic Server*.
- Make sure the Decision Manager JVM trust store trusts the certificate authority that Oracle RTD Server uses.

> **Note:** Use a real certificate and not the one in the RTD demo trust store. Configure Decision Studio, Load Generator and other RTD clients to use the right trust store by specifying the `javax.net.ssl.trustStore` as a Java command line argument.

- Make sure the Decision Manager JVM trust store trusts the certificate authority that Oracle BIEE Server uses.

Note: In case the OBIEE server key is 512 bits only, the Decision Manager JVM may reject it for being too short. Either change the size of the OBIEE server key, or specify in the Decision Manager JVM that 512 bit keys are acceptable.

- See “Default x.509 Certificates Have Longer Key Length” in the Java™ SE Development Kit 7, Update 40 (JDK 7u40) Release Notes.
- Use HTTPS instead of HTTP in the Inline Service configuration (either in the configuration file, see Section 2.2.2.1, "Config XML File" or in the Inline Service Configuration dialog, see Section A.12.1, "Inline Service Configuration" in the *Oracle® Real-Time Decisions Base Application Decision Management Applications User's Guide*).
- Use HTTPS instead of HTTP in the BI Server configuration (in `<bi-server>` the configuration file, see Section 2.2.2.1, "Config XML File").

2

Configuring Oracle RTD Decision Management

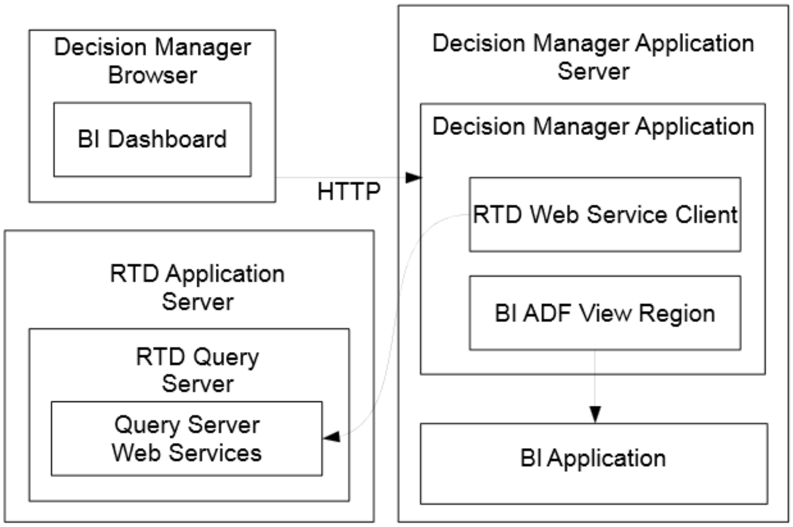
Terminology: The term *reference implementation* is used in this chapter to refer to the specific Oracle RTD Decision Management application RTD for Marketing Optimization, released with Oracle RTD Base Application.

This chapter contains the following topics:

- Section 2.1, "Oracle RTD Decision Management Architecture Overview"
- Section 2.2, "Configuring Oracle RTD Decision Management"
- Section 2.3, "Configuring and Deploying Base Marketing with Slices"
- Section 2.4, "Configuring Logs"

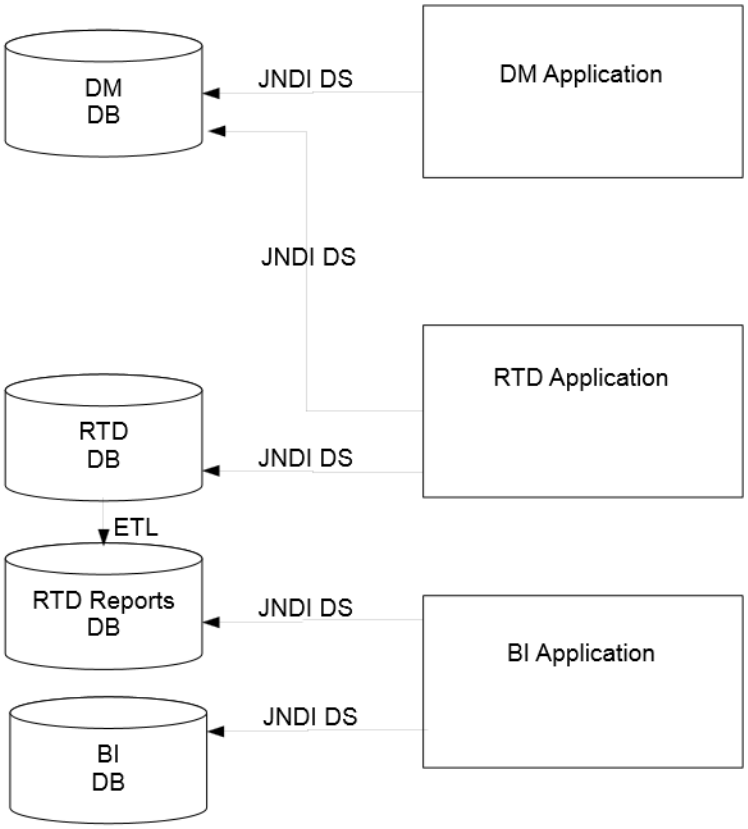
2.1 Oracle RTD Decision Management Architecture Overview

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



The Oracle RTD Decision Manager application is an ADF based web application. The Oracle RTD Decision Manager application makes web service (SOAP) calls to the Oracle RTD workbench service and discovery explorer service running on one of the Oracle RTD Query Servers. These calls are used to retrieve type restrictions, display the rule editor and the model reports. These requests use the username and password that you configured in Section Section 1.2.3.4, "Storing Credentials to Enable Web Service Calls" for WebLogic.

The following diagram shows the communications at runtime between the various components of Oracle RTD Decision Management and the databases they use:



The Oracle RTD Decision Management application accesses the Decision Management database using the credentials defined in the JNDI datasource specified in Section 1.2.3.3, "Oracle RTD Decision Management Data Source Setup".

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.

An ETL process transform data from the Decision Analytics tables in the Oracle RTD (SDDS) database into a star schema in the RTD Reports database.

The BI application access its own database schema and reports on the information in the RTD Reports database.

2.2 Configuring Oracle RTD Decision Management

This section contains the following topics:

- Section 2.2.1, "Overview"
- Section 2.2.2, "Oracle RTD Decision Management Metadata Configuration Files"
- Section 2.2.3, "Install and Configure Oracle JDeveloper"

- Section 2.2.4, "Ant Tasks"
- Section 2.2.5, "Java API"
- Section 2.2.6, "Application Extensions"

2.2.1 Overview

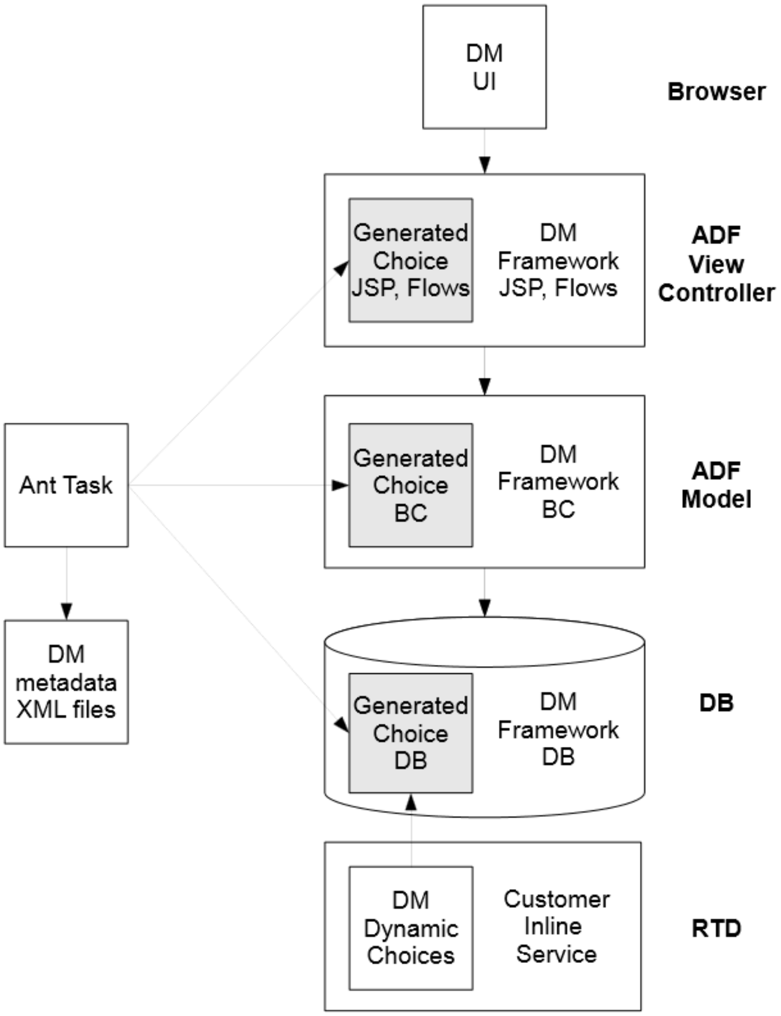
Oracle RTD Decision Management applications are built and deployed using Decision Designer.

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

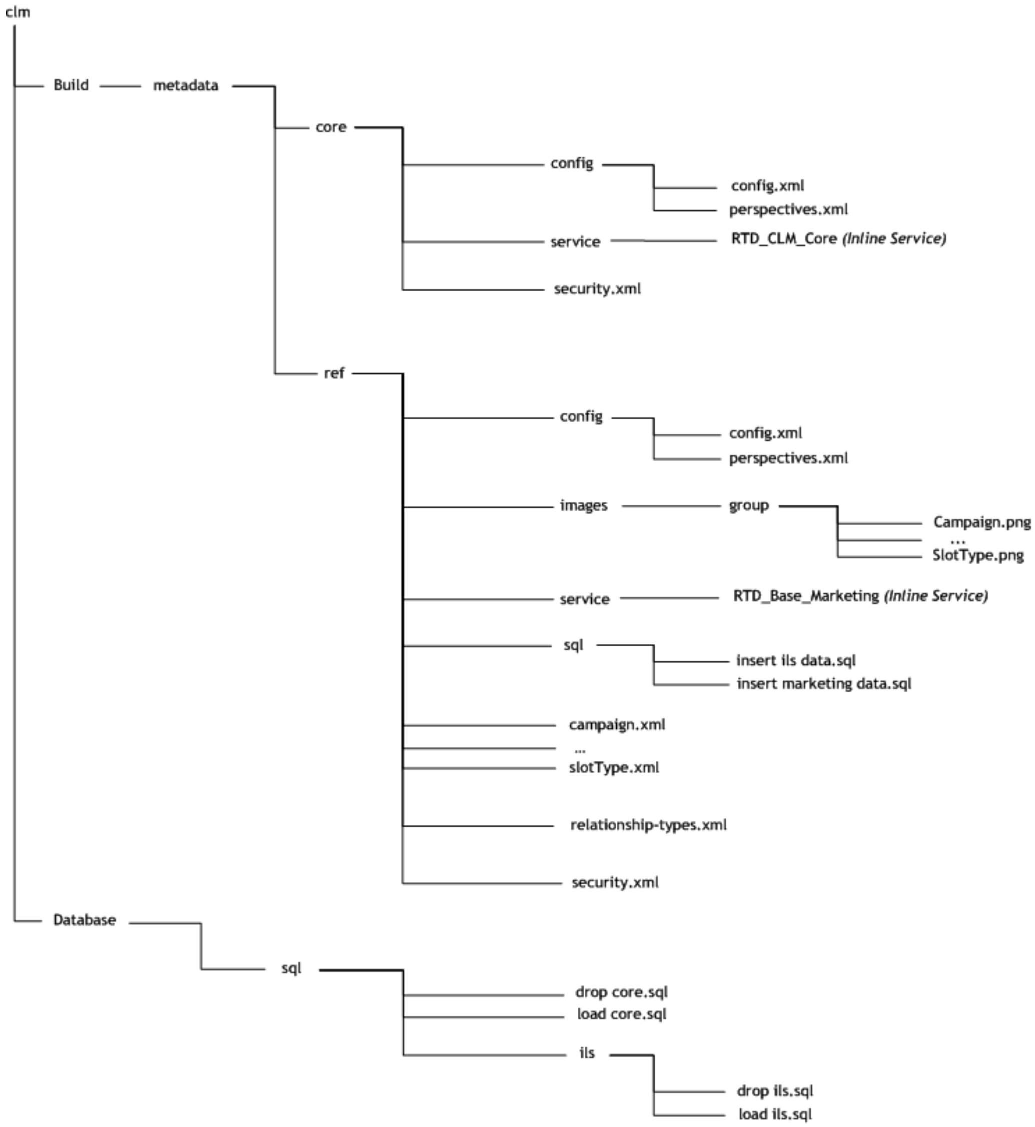
The following diagram shows the components that are generated by Decision Designer.



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 Manager 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.

The folder `clm\Build\metadata\ref-slices` contains the metadata for the reference implementation with the addition of the slices feature; for details, see Section 2.3, "Configuring and Deploying Base Marketing with Slices".

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.

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

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 "&lt;" (instead of "<") and "&gt;" (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 and must be specified in this order:

- enterpriseAppName (optional) specifies the application name used when multiple applications are deployed into the same domain. The default value is clm.
- webAppContextRoot (optional) 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`.
- datasourceJNDIName (optional) specifies the data source name used when multiple applications are deployed into the same domain. The default value is CLMDS.
- inlineService (required) specifies the Inline Services used. Multiple inline services can be inserted. Each inline service has the following attributes:
  - url: the URL to the RTD server (using HTTP or HTTPS)
  - inlineServiceName: the name of the Inline Service
  - version: must be RTD3
  - deploymentState: must be 5
  - default: one of them must be true. This is the one used for type restrictions, performance goals, segments and the rule editor. The other ones can be used for reports and are optional.
- ownership-mode (optional) 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.

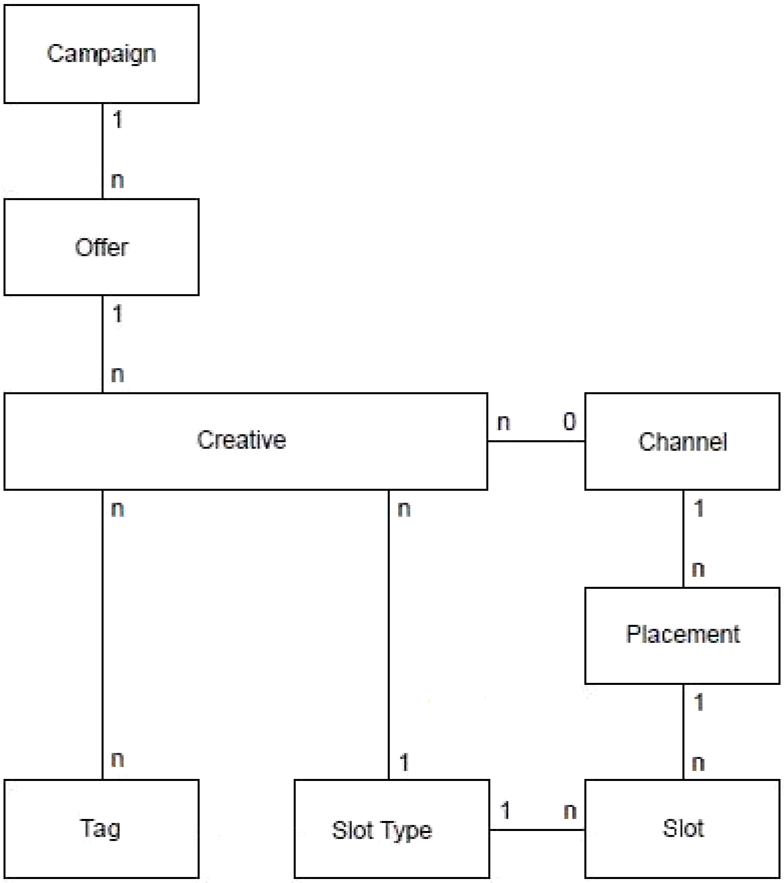
- display-name (optional): specifies the name of the application as it appears on the login page and at the top of the main page.
- database-encoding (optional): The value of database-encoding depends on the database character set. Oracle RTD Decision Manager uses database encoding to properly enforce 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`.
- bi-server (optional): specifies the OBIEE server to populate Decision Analytics Dashboard in a tab that contains a bi-dashboard parameter. If you do not have a bi-server in `config.xml`, Any tab that contains a bi-dashboard will be hidden. bi-server has two attributes:
  - `url` specifies the URL (http or https) to the OBIEE server; for example, `url="http://`*`<server>`*`:`*`<port>`*`/analytics"`.
  - `wsdl-context` specifies the WSDL used; for example, `wsdl-context="analytics-ws"`.

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

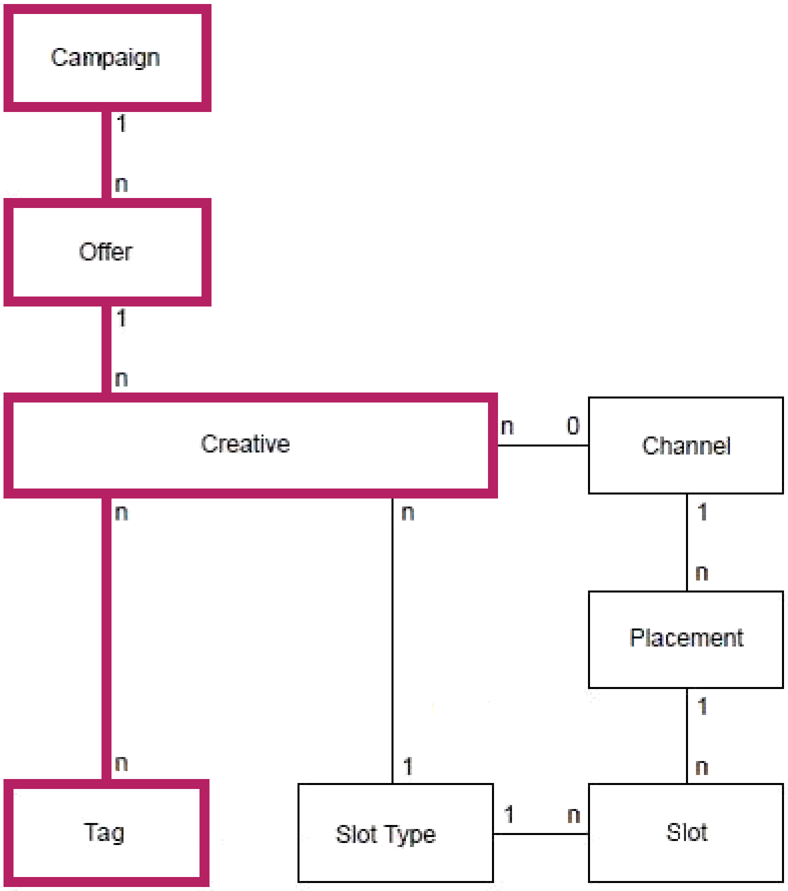


Figure 2–3 Tags Perspective

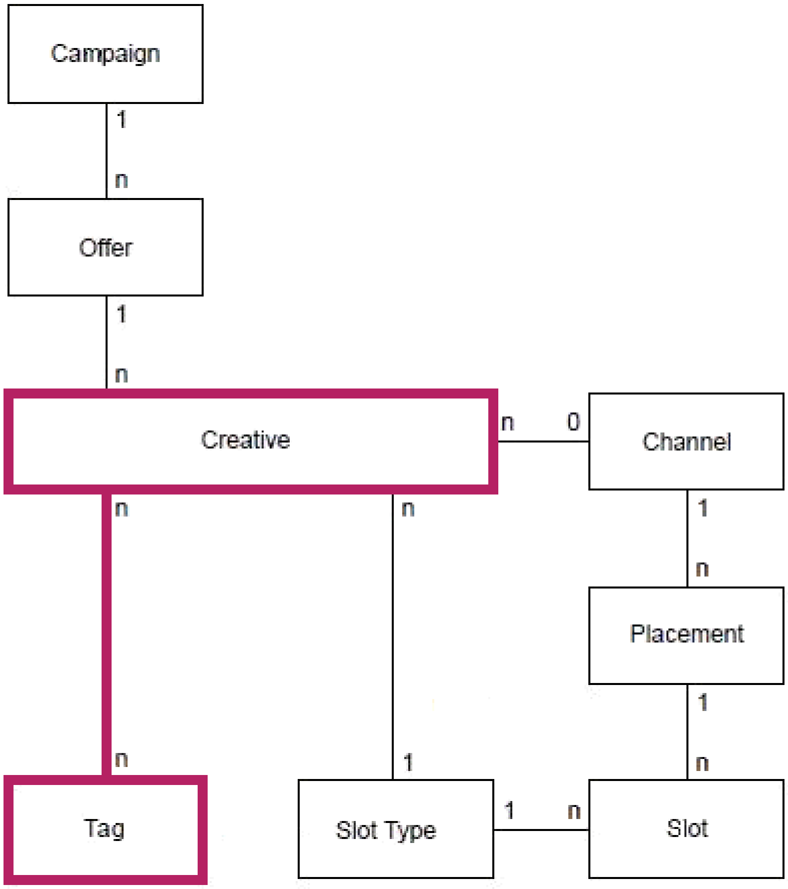


Figure 2–4 Slot Types Perspective

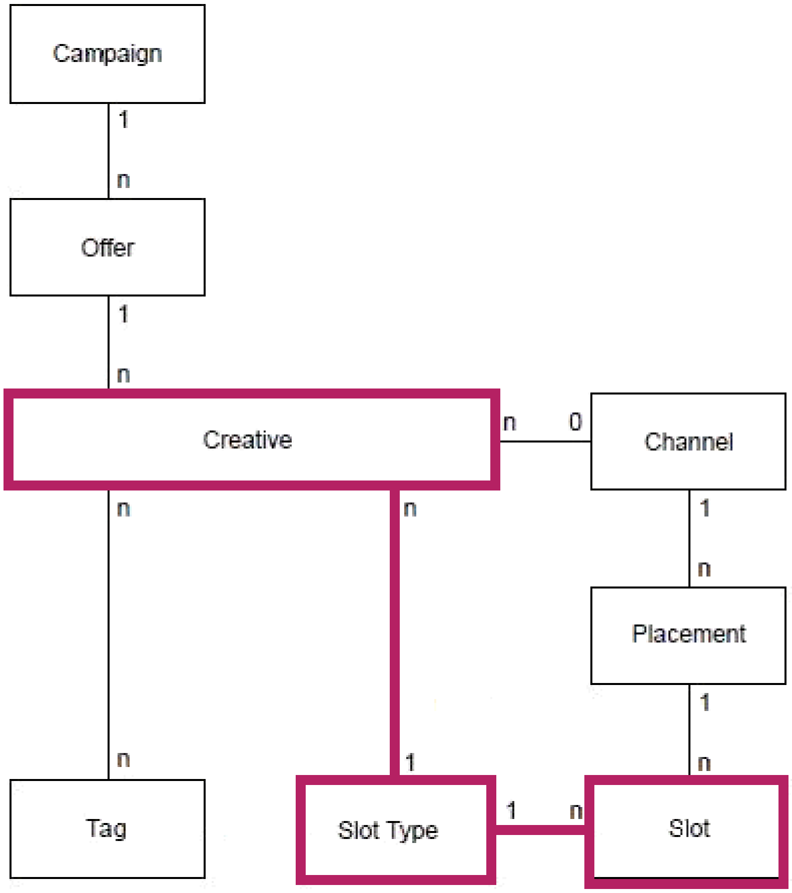
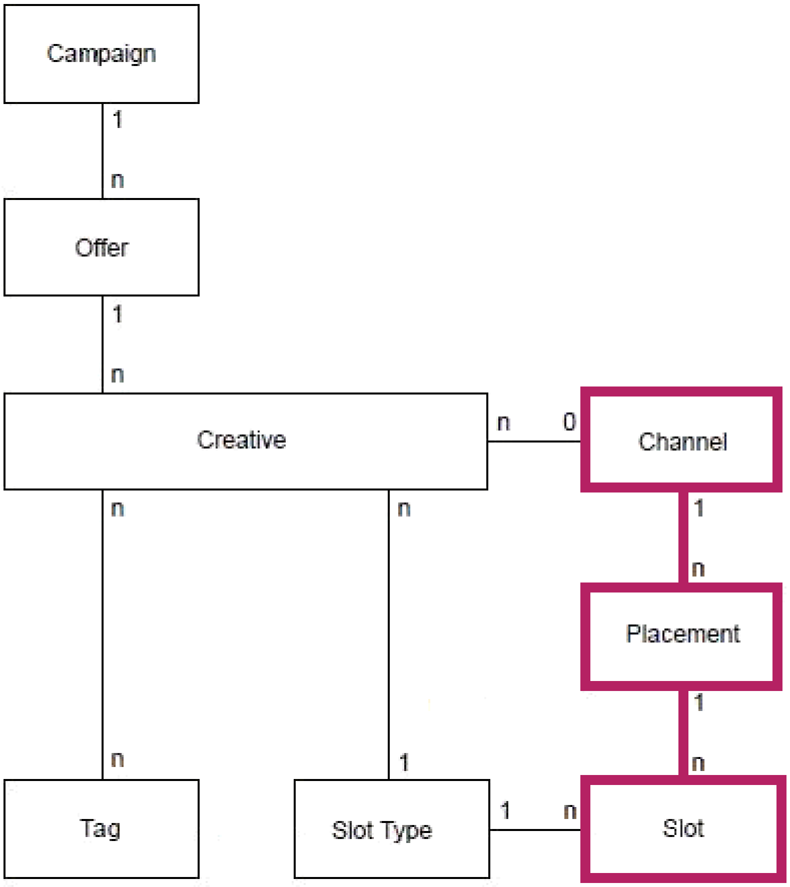


Figure 2–5 Channels Perspective



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.

Properties of perspective:

- name (required): the name of the perspective
- description (optional)

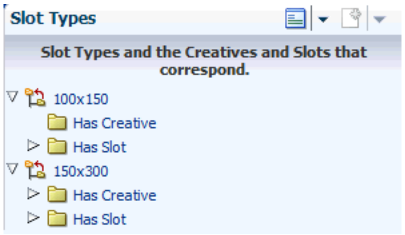
  If present in perspectives.xml, the description appears as the heading for the perspective in the left-hand Perspective panel of the Decision Manager user interface. It also provides the tool tip help information for Decision Manager users as they mouse hover over the perspective name either in the Perspective 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 (required)
- multiple level nodes (ordered; optional)

  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 then Creative, Slot-Type then 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`):



Slot Types

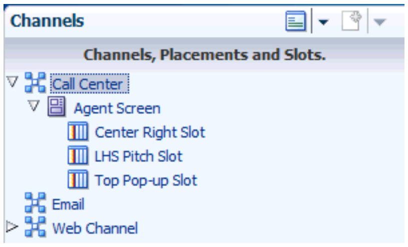
Slot Types and the Creatives and Slots that correspond.

- 100x150
  - Has Creative
  - Has Slot
- 150x300
  - Has Creative
  - Has Slot

On the other hand, a hierarchical association Channel then Placement then 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:



Channels

Channels, Placements and Slots.

- Call Center
  - Agent Screen
    - Center Right Slot
    - LHS Pitch Slot
    - Top Pop-up Slot
- Email
- Web Channel

Properties of root:

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

Properties of level:

- choiceGroupId (required) the choice group that will be the node of the tree at that level
- relationshipTypeId (required): the relationship to follow between the previous level and this level
- reversed (optional): 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 (optional)

  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 (optional; see later in this section)
- viewCriteria (optional; see later in this section)
- simpleViewCriteria and viewCriteria (both optional) 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 (optional): 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 Manager 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 (required): the id of the choice group. It must match the id in the Inline Service
- name (required): the name of the choice group. It will appear as such in the Decision Manager user interface
- description (optional): the description of the choice group.
- searchSortOrder (optional): 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 (optional): 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 (optional): an array of attribute (see later in this section)
- criteria (optional): zero to many criteria used for perspective filtering (see later in this section)

Properties of attributes:

- id (required): the id of the choice attribute. It must match the id in the Inline Service.
- name (required): the name of the choice attribute. It will appear as such in the Decision Manager user interface.
- description (optional): 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: (optional): 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:

  - "&lt;&gt;" (Not equal to)
  - "NOTBETWEEN" (Not between)
  - "CONTAINS" (Contains)
  - "ISNOTBLANK" (Is not blank)
  - "=" (Equal to)

- "&lt;" (Less than)
- "&gt;" (Greater than),
- "ISBLANK" (Is blank)
- "&lt;=" (Less than or equal to)
- "&gt;=" (Greater than or equal to)
- "DOESNOTCONTAIN" (Does not contain)
- "STARTSWITH" (Starts with),
- "ENDSWITH" (End with)
- "BETWEEN" (Between)

Valid values for dates:

- "&lt;&gt;" (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:

- "&lt;&gt;" (Not equal to)
- "ISBLANK" (Is blank)
- "&gt;" (Greater than)
- "&lt;=" (Less than or equal to)
- "&gt;=" (Greater than or equal to)
- "NOTBETWEEN" (Not between)
- "ISNOTBLANK" (Is not blank)
- "BETWEEN" (Between)
- "&lt;" (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 "&lt;" and the "greater than" operator is written as "&gt;".

For example:

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

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

Properties of type:

- name (required): the type of the attribute, either "string", "date", "number", or "clob".
- length (optional): 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 (optional): 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 (optional): 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 (optional): 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 (optional): whether a non-null, non-empty value must be entered.

Properties of restriction:

- kind (required): 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 (optional): for range type restrictions, whether the range has a min bound ("true" or "false")
- maxDefined (optional): for range type restrictions, whether the range has a max bound ("true" or "false")
- minInclusive (optional): for range type restrictions, whether the range min bound is included ("true" or "false")
- maxInclusive (optional): 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 (required): 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: (optional) 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 (optional): `<default-value value="adf.currentDate" expression="true"...... />`

- masterValue (optional): 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 (required): the widget name, which must be one of the following:
  - InputDate - this widget creates a text field for entering dates and a glyph which opens a popup for picking dates from a calendar.
  - 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).
- PerformanceGoalsEditor - This creates a widget which helps in assigning weights to the Performance Goals. This widget uses these attributes:
  - inlineService (optional): specifies to which inline service is being used.
  - ruleChoiceGroupId (optional): The name of the choice group whose segments should be displayed in the PerformanceGoalsEditor widget. This is used when the name of the control is set to PerformanceGoalsEditor.
  - includeIlsSegments (optional): It can be set to `true` or `false`. This indicates whether static segments should be shown in the PerformanceGoalsEditor widget. This is used when the name of the control is set to PerformanceGoalsEditor.
- SelectOneChoiceLOV - this widget displays list of values derived from a Type Restriction.
- Image - this widget displays 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)*.
- InputText - This widget is used to enter values for attributes. An InputText widget can have one optional attribute, rows. rows determines if a text field will have one or more rows. If rows is not specified, the default value is 1
- RTDRuleEditor - the 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 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.
  - inlineService

    Specifies the Inline Service used by this rule. If not specified, the default inline service is used.
  - object, type and editingAspect

This type shows what to enter for these attributes based on the type of rule you want:

Rule Type	object	type	editingAspect
Choice Eligibility Rule	Choice Group Id (Optional; this defaults to the choice group ID of the choice group this rule is in)	choice (Optional; this is the default value)	rule (Optional; this is the default value)
Filtering Rule	*(Omit this parameter)*	eligibilityRule	whole
Scoring Rule	*(Omit this parameter)*	valueRule	rule

Properties of relationship types:

- id (required): the id of the relationship type. You use this id in your Inline Service for methods such as CLMChoiceBag.getRelatedChoiceIds().
- fromChoiceGroupId (required): the choice group source.
- toChoiceGroupId (required): the choice group destination.
- fromName (required): the owner to destination relationship type name. You use this name in your Inline Service for methods such as CLMChoiceBag.getRelatedChoiceIds().
- toName (required): the destination to owner relationship type name. You use this name in your Inline Service for methods such as CLMChoiceBag.getRelatedChoiceIds().
- fromDescription (optional): the owner to destination relationship type description.
- toDescription (optional): the destination to owner relationship description.
- cardinality (required): 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 (optional): '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 (optional): '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 (optional): '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 9.2.11, "au" in *Oracle Real-Time Decisions Base Application Installation and Reference Guide*

Properties of criteria:

- id (required): the id of the criteria
- xml (required): 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 (optional): top-level tag that is the container for the Decision Manager user interface layout.
- choice-view (optional; within views): the Decision Manager 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.

- tab (optional; within choice-view): A tab contains one or more pages as its elements. It contains the attributes description, `id`, `name`, and `type` as its attributes. The `type` attribute can be either *choice* or *report*. The following example shows a tab with the type attribute of choice:

```
<tab id="Definition" type="choice">
   <name>Definition</name>
   <description>Definition</description>
   <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>
        <note viewType="edit"/>
      </page-attributes-group>
    </page>
   <page id="Attributes">
   <name>Attributes</name>
   <description>Attributes</description>
      <page-attributes-group>
         <page-attributes-group>
            <attribute id="approvalStatus"/>
            <attribute id="code"/>
            <attribute id="startDate"/>
            <attribute id="endDate"/>
            <attribute id="type"/>
            <attribute id="region"/>
         </page-attributes-group>
         <note viewType="edit"/>
       </page-attributes-group>
    </page>
</tab>
```

- page (optional; within tab): The page 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 below).

  A page corresponds to either a tab (displayed when choices and projects are edited or viewed) or a train stop (displayed during choice creation). This tab of train stop is embedded under the top level tabs defined by tab metadata (see above).

  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 (optional; 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.

- model-report (optional; within page-attributes-group): Within the Decision Manager metadata you can specify integrating a model report within a tab. The syntax is `<model-report reportType="<type>"/>`

  Where type is one of:

  - `choiceHistory`; that is, Choice Performance Model Counts
  - `predReport`; that is, Choice Analysis Predictiveness
  - `bestFit`; that is, Choice Analysis Best Fit
  - `drivers`; that is, Choice Analysis Drivers
  - `modelQuality`; that is, Choice Analysis Model Quality

- bi-dashboard (optional; within page-attributes-group): Within the Decision Manager metadata you can specify integrating integrating a BI dashboard within a tab. The syntax is:

  ```
  <bi-dashboard path="<path>">
     <parameter name="<name>" value="<value>"/>
     …
  </bi-dashboard>
  ```

  `<path>` is the path to the dashboard in the web catalog, for instance `"/<shared/Decision Analytics - Base Marketing/Creatives/_ portal/Creative Performance Dashboard">` and it supports multiple parameter name/value pairs.

  The following example shows how to code a Campaign choice group in the Campaign Performance Dashboard.

  ```
  <bi-dashboard path="/shared/Decision Analytics - Base Marketing/Campaigns/_
  portal/Campaign Performance Dashboard">
  ```

```
   <parameter name='Applications."App Name"'
value="#{biParameters.serviceName}"/>
   <parameter name="Choices.HIERARCHY_ID_1" value="#{biParameters.choiceId}"/>
   <parameter name="dashboard.variables['DMFrequency']"
value="#{biParameters.timeInterval}"/>
   <parameter name='Time."Calendar Date"' operator="between"
value="#{biParameters.localizedStartDateString},#{biParameters.localizedEndDate
String}" />
</bi-dashboard>
```

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

  May contain the readonly and viewType sub-tags.

  All relationships for view`Type = "view"` are readonly.

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

- note (optional; 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 (optional; within page-attributes-group): appears as text message on the page.
- audit-trail (optional; within page-attributes-group): renders the Audit Trail for this choice.

2.2.2.4.1 Example of User interface XML for Placement Choice Group `<views>`

```
<choice-view choiceGroupId="Placement" type="create edit view">
  <tab id="Definition" type="choice">
    <name>Definition</name>
    <description>Definition</description>
    <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>
          <relationship id="Placement to Channel"/>
        </page-attributes-group>
        <note viewType="edit"/>
```

```
          </page-attributes-group>
        </page>
        <page id="Attributes">
          <name>Attributes</name>
          <description>Attributes</description>
          <page-attributes-group>
            <page-attributes-group>
              <attribute id="type"/>
              <attribute id="url"/>
            </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"/>
          <page-attributes-group>
            <note viewType="edit"/>
          </page-attributes-group>
          </page-attributes-group>
        </page>
        <page id="Confirm" viewType="create">
          <name>Confirm</name>
          <description>Confirm</description>
          <page-attributes-group>
            <text value="#{nls['ID_EXPLAIN_VALUE']}"/>
            <text value="#{nls['ID_EXPLAIN_ALLOWED']}"/>
            <attribute id="ChoiceId" readonly="false"/>
            <note/>
            <page-attributes-group>
              <text value="#{nls['CONFIRM_EXPLAIN_VALUE']}"/>
              <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>
            <attribute id="EligibilityRule" readonly="true"/>
          </page-attributes-group>
        </page>
      </tab>
      <tab id="Performance" type="report">
        <name>Performance</name>
        <description>Performance</description>
        <page id="PerformanceDashboard" viewType="view">
          <name>Dashboard</name>
          <description>Dashboard</description>
          <page-attributes-group>
            <bi-dashboard path="/shared/Decision Analytics - Base
Marketing/Placements/_portal/Placement Performance Dashboard">
              <parameter name='Applications."App Name"'
                         value="#{biParameters.serviceName}"/>
              <parameter name="Choices.HIERARCHY_ID_4"
                         value="#{biParameters.choiceId}"/>
              <parameter name="dashboard.variables['DMFrequency']"
                        value="#{biParameters.timeInterval}"/>
              <parameter name='Time."Calendar Date"' operator="between"

value="#{biParameters.localizedStartDateString},#{biParameters.localizedEndDateStr
ing}" />
            </bi-dashboard>
          </page-attributes-group>
        </page>
        <page id="ChoiceHistory" viewType="view">
          <name>Model Counts</name>
          <description>Model Counts</description>
          <page-attributes-group>
            <model-report reportType="choiceHistory"/>
          </page-attributes-group>
        </page>
      </tab>
      <tab id="Analysis" type="report">
        <name>Analysis</name>
        <description>Analysis</description>
        <page id="PredReport" viewType="view">
          <name>Predictiveness</name>
          <description>Predictiveness</description>
          <page-attributes-group>
            <model-report reportType="predReport"/>
          </page-attributes-group>
        </page>
        <page id="BestFit" viewType="view">
          <name>Best Fit</name>
          <description>Best Fit</description>
          <page-attributes-group>
            <model-report reportType="bestFit"/>
          </page-attributes-group>
        </page>
        <page id="Drivers" viewType="view">
          <name>Drivers</name>
          <description>Drivers</description>
          <page-attributes-group>
            <model-report reportType="drivers"/>
          </page-attributes-group>
        </page>
        <page id="ModelQuality" viewType="view">
          <name>Model Quality</name>
          <description>Model Quality</description>
          <page-attributes-group>
            <model-report reportType="modelQuality"/>
          </page-attributes-group>
        </page>
      </tab>
      <tab id="Audit" type="report">
        <name>Audit Trail</name>
        <description>Audit Trail</description>
        <page id="AuditTrail" viewType="view">
          <name>Audit Trail</name>
          <description>Audit Trail</description>
          <page-attributes-group>
            <audit-trail/>
          </page-attributes-group>
        </page>
      </tab>
    </choice-view>
  </views>
```

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.
- `<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 children, 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>
```

For the configuration resource type:

- This permission, to be used only by admin users, is for accessing the Inline Service Configuration dialog. This dialog lets administrators specify how to connect to RTD for retrieving type restrictions, displaying the rule editor and displaying reports.

  Example:

  ```
  <permission>
    <resource-type>configuration</resource-type>
    <resource-name>ils-configuration</resource-name>
    <actions>create,read,update,delete</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 Install and Configure Oracle JDeveloper

To compile and create an EAR file for your Decision Manager applications, you must install JDeveloper 11g Release 1. Once installed, do the following:

1. Open JDeveloper and
   a. Select Tools, then Preferences.
   b. Select the Environment tab and set the Encoding to UTF-8.

   c. Click **OK**.

2. Create an `ANT_HOME` and a `JAVA_HOME` environment variable pointing to the location where ant and the Java JDK are installed, for instance:

   ```
   ANT_HOME=<Oracle JDeveloper Install dir>\Middleware\jdeveloper\ant
   JAVA_HOME=<path to Java JDK>
   ```

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

4. 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 JDeveloper Install dir>/Middleware
   ```

   For example, `C:/JDeveloper/Middleware`

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

2.2.4 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`.

  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 Manager templates to extend the behavior of the application without modifying any of the core and generated files.

2.2.5 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`.

2.2.6 Application Extensions

This section contains the following topics:

- Section 2.2.6.1, "Adding an Attribute to a Choice Group or Project"
- Section 2.2.6.2, "Adding a Choice Group"
- Section 2.2.6.3, "Creating a Relationship Type Between Two Choice Groups"
- Section 2.2.6.4, "Modifying Perspectives"

2.2.6.1 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.

2.2.6.2 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.

2.2.6.3 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.

2.2.6.4 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 Configuring and Deploying Base Marketing with Slices

This section shows you how to configure and deploy Base Marketing with Slices. It contains these topics:

- Section 2.3.1, "Preparing Base Marketing with Slices Application for Deployment"
- Section 2.3.2, "Build and Generate the EAR"
- Section 2.3.3, "Prepare the SQL Scripts for Database Setup"
- Section 2.3.4, "Install Oracle RTD Decision Manager Slices Application"
- Section 2.3.5, "Configuring Security"

- Section 2.3.6, "Mapping Users or Groups to Slice-based Roles"

2.3.1 Preparing Base Marketing with Slices Application for Deployment

To generate the `clm.ear` file for the Base Marketing With Slices application, perform the following tasks:

1. Update Oracle RTD Decision Manager Metadata Configuration Files
2. Update Relationship-Types and Relationship-Slices XML Files for Choice Groups
3. Understand User Interface Pages
4. Understand Security XML Files
5. Understand the Folder for Role and Slice Choice Group Images
6. Understand the Slices Inline Service Folder

2.3.1.1 Update Oracle RTD Decision Manager Metadata Configuration Files

This sections explains the changes made in metadata and its files for Slices feature.

To use Slices, you need to define in the metadata:

- Slice and Role choice groups
- Relationships between roles, slices and project choice groups, as declared in `relationship-slices.xml` from ref-slices metadata.

Relationships between all other choice groups and slices are generated automatically during code generation.

2.3.1.1.1 Config XML file The attributes and its description remain the same as described in Section 2.2.2.1, "Config XML File" except that the values or default values for some of the attributes are those described here:

- inlineService: The default value of the inlineServiceName attribute would be `RTD_Base_Marketing_Slices`.
- datasourceJNDIName: The default value would be `DM_SLICES`. Modify it if datasource is created with different name.
- enterpriseAppName: The default value is `RTD Base Marketing (with slices).`
- webAppContextRoot: The default value is `/dmsl`. Modify it if necessary and ensure that the modified value is used when trying to access the Slices URL.

2.3.1.1.2 Perspectives XML File In addition to perspectives explained in Section 2.2.2.2, "Perspectives XML File", Slices requires two additional perspectives:

- Roles
- Slices

The Roles and Slices perspectives is having Single root node with no multiple levels. They are defined in `perspectives.xml` as shown here:

```
<perspective name="Roles">
   <description>Roles</description>
   <root choiceGroupId="Role">
   </root>
</perspective>
<perspective name="Slices">
   <description>Slices</description>
   <root choiceGroupId="Slice">
   </root>
</perspective>
<perspective name="Segments">
   <description>Segments</description>
   <root choiceGroupId="Segment" />
</perspective>
<perspective name="Decisions">
   <description>Decisions</description>
   <root choiceGroupId="Decision" />
</perspective>
```

2.3.1.2 Update Relationship-Types and Relationship-Slices XML Files for Choice Groups

This section describes the new Choice Groups,“Role” and “Slice”, added for Slices-based marketing application and the additions and modifications they require in the relationships metadata file.

2.3.1.2.1 Choice Group XML Files In addition to the existing choice groups used for Base Marketing application, the Slices feature introduces two new choice groups:

- Role: `role.xml` (`ref-slices/` folder)
- Slice: `slice.xml` (`ref-slices/` folder)

The following table describes the properties and default values for the Role and Slice Choice groups:

Property	Role	Slice	Description
id	Role	Slice	
name	Role	Slice	
description	A Role	A slice	
searchSortOrder	1011	1012	
createSortOrder	1011	1012	

Unlike other Choice Groups, for example Campaign or Creative, the Role and Slice Choice Groups have no attributes.

2.3.1.2.2 Relationship-Slices XML Files The new file `relationship-slices.xml` has been added to define relationship between Project, Role and Slice Choice Groups.

For operations, Roles have relationship to slices:

- **Role to Slice Create**: Slices that should be assigned by default for newly created choices.
- **Role to Slice Read**: Slices assigned to choices that can be read but not changed.
- **Role to Slice Write**: Slices assigned to choices that can be read and changed.

> **Note:** Names of these relationships cannot be changed.

For operations, Projects are assigned to roles as follows:

- **Project to Role Read**: Project is read-only for users who have specified roles.

- **Project to Role Write**: Project can be changed by users who have specified roles.

> **Note:** Names of these relationships cannot be changed.

You should have at least one slice-related role to log in to an application. Some database initial script should be used to create an administrator user and "slice-related role" in application ("ref-slices" metadata contains script "insert ils data.sql"). Also, if some choices created via database initial script, they should be linked to some slices, because choices without slices can't be accessible via application.

After that, administrator will be able to create additional "slice-related" roles and assign slices to choices.

2.3.1.2.3 Relationship-Types XML File There is no change to this file for Slices based Marketing application as the relationship between other choice groups like Campaign, Creative etc and Slice choice groups would be generated automatically during code generation phase.

2.3.1.3 Understand User Interface Pages

The layout of the pages and tabs used when creating, editing, and viewing choices and projects is determined by XML metadata defined in `role.xml`, `slice.xml` and `project.xml`.

Key changes you should note are:

- `project.xml`:
  - **Project to Role Read** relationship is defined within the `page-attributes-group` on the Overview page.
  - **Project to Role Write** relationship is defined within the `page-attributes-group` on the Overview page
- `role.xml`:
  - **Role to Slice Read** relationship is defined within the `page-attributes-group` for the Slices and Confirm pages.
  - **Role to Slice Write** relationship is defined within the `page-attributes-group` for the Slices and Confirm pages.
  - **Role to Slice Create** relationship is defined within the `page-attributes-group` for the Slices and Confirm pages.
- `slice.xml`:

  By default, `slice.xml` has two pages, Overview and Confirm. Since its page does not have to display any other choice group information, the relationship is not defined here.

2.3.1.4 Understand Security XML Files

By default, all security metadata is in `security.xml`. For more information, see Section 2.2.2.5, "Security XML Files".

2.3.1.5 Understand the Folder for Role and Slice Choice Group Images

The `Role.png` and `Slice.png` image files are stored in `<`*`metadata_modules_home`*`>\ref-slices\images\group`.

2.3.1.6 Understand the Slices Inline Service Folder

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

2.3.2 Build and Generate the EAR

If you have made changes to the configuration:

1. From the `clm\Build` directory, run this command:

   ```
   ant generate -Dmetadata.module=ref-slices
   ```

2. Then run this command:

   ```
   ant ear
   ```

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

2.3.3 Prepare the SQL Scripts for Database Setup

To prepare the SQL scripts for the database, do the following:

1. Create the database user by following the steps in Section 1.2.3.2, "Oracle RTD Decision Management Database Creation".
2. Execute load `core.sql` and load `ils.sql` to set up the production database. These files are located in `clm\Database\sql` and `clm\Database\sql\ils`. respectively.
3. Provide the Inline Service. It should be located in `clm\Build\metadata\ref-slices\service`.

2.3.4 Install Oracle RTD Decision Manager Slices Application

This section describes how to install Oracle RTD Decision Manager Slices. This task comprises these steps:

1. Configure and Set Up the Application Server
2. Set Up the Oracle RTD Decision Manager Data Source for Slices
3. Deploy the Inline Service
4. Deploy Oracle RTD Decision Manager Slices

2.3.4.1 Configure and Set Up the Application Server

Configure and set up the application server as described in Chapter 1, "Installing Oracle RTD Decision Management".

2.3.4.2 Set Up the Oracle RTD Decision Manager Data Source for Slices

Create a data source with following properties:

- **Name:** DM_SLICES_DS or a similar value
- **JNDI Name:** The proper JNDI name.

  The JNDI name must be the JNDI name you specify in `config.xml` and in your Inline Service application parameter; by default, this is DM_SLICES

- **Database Type:** Oracle

> **Note:** Ensure that the Database User name and Password match the values that you set up in Section 2.3.3, "Prepare the SQL Scripts for Database Setup".

2.3.4.3 Deploy the Inline Service

Once deployed, RTD Slices requires its Inline Service to be running in the Oracle RTD instance. You should have received this Inline Service with the clm.ear file or it should be located in `clm\Build\metadata\ref-slices\service`.

Using Decision Studio, deploy this Slices Inline Service to the Oracle RTD instance that is configured in config.xml or in any other RTD instance that can be accessed through Decision Manager Slices application.

2.3.4.4 Deploy Oracle RTD Decision Manager Slices

To deploy Oracle RTD Decision Manager Slices, do the following:

1. Ensure that the credentials used to access Oracle RTD from Oracle RTD Decision Manager are set up, as described in Section 1.2.3.4, "Storing Credentials to Enable Web Service Calls".
2. Deploy the Oracle RTD Decision Manager. Use the EAR either generated in Section 2.3.2, "Build and Generate the EAR".
3. Deploy the EAR for your application server.

The deployed Oracle RTD Decision Manager Slices application is now accessible at the URL `http://`*`<server>`*`:`*`<port>`*`/dmsl`.

The URL depends the value provided in `config.xml` for `<webAppContextRoot>`. By default `dmsl` is used as context root, as shown here.

```
<webAppContextRoot>/dmsl</webAppContextRoot>
```

If necessary, you can change `dmsl` to a different value and access URL by using that name.

2.3.5 Configuring Security

In addition to the security configuration defined in Section 1.4, "Security Configuration", Slices provides an additional security layer by introducing new Role called Slice-based Role. In order to view and apply Slice based permission to choices, a user needs to have Slice-based Role in addition to standard Roles and configuration defined in Section 1.6.

This section describes the tasks necessary to configure security for your Oracle RTD Slices implementation:

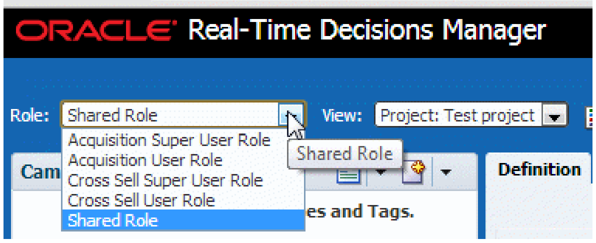
1. Define Slice-based Roles in security.xml
2. Create an Application Role (Standard Role) with Permission for Role and Slice Choice Groups in security.xml
3. Create an Application Role for a Slice-based Role in security.xml
4. Generate the Decision Manager Application
5. Deploy RTD Decision Manager

2.3.5.1 Define Slice-based Roles in security.xml

The choice group level permissions and Slice based permission together determine the actions allowed to a user.

To access choices in a Slice-based application, the application role should have at least one of the CRUD (Create, Read, Update or Delete) permissions on the Role and Choice groups, in addition to the Slice-based Role permission.

You work with Decision Manager within the scope of one of the Slice-based roles to which he or she belongs. The topmost drop-down in the user interface is a role selector with which you select a role from those to which you belong:



Role: Shared Role

View: Project: Test project

The current role chosen from the drop-down and the choice group, page, and perspective permissions act as a filter to restrict what you can do while acting in that role. Some examples are:

- A user could be an administrator with CRUD permission on all choice groups but when acting in a specific role, you would only be able to perform the actions allowed by that role.
- A user with few permissions could act in a role that allows many actions but might still be restricted to the subset for which he or she already has permission.
- Some pages might be only for roles that, for instance, allow a user to hide some attributes or make them read-only for some roles even if the user has permission to edit that attribute in other roles.
- Some perspectives can be made available only for some roles, which are restricted to the perspectives only a given role can see; for example, no administration perspective for someone with a basic user role.

  For example, if a user is mapped to the roles CLMSliceAdministrator and Cross Sell User Role and Cross Sell User Role has been selected, that user can only Create, Edit or View Cross Sell slice based choices but cannot view Acquisition slice based choices, even though the user is an Administrator.

`security.xml` contains the application roles that can be assigned to a user through Enterprise Manager or an application server, such as WebLogic.

The application roles defined in `security.xml` contains a set of permissions on the Choice Group, View Page, and so on that determine which privileges of a user or a group linked to these application roles.

RTD Slices enables users to create a new set of Slice-based application roles, with the user can define a Slice-based permission on choices in addition to a Choice Group permission.

The Slice-based Role—that is, choices of type Role choice group—are available for assignment to users through EM if the role is defined in `security.xml` with the same name used in the CLM database.

2.3.5.2 Create an Application Role (Standard Role) with Permission for Role and Slice Choice Groups in security.xml

To create an application role, do the following:

1. In `clm/Build/metadata/ref-slices`, edit `security.xml` or create a different file, such as `groups.xml`.
2. Create an application Role in `security.xml` that has either a choice group permission on individual choice groups like Role, Slice, and other choice groups or on _all_ (all choice groups). This ensures that users mapped to this role have permission to access respective choices. See the following example cases:

To create an application role having CRUD permission on Campaign, Slice, Role, and Offer choice group, you would enter the following in `security.xml`:

Example Case 1:

```
<application-role>
   <name>CLMSliceManager</name>
   <permissions>
      <permission>
         <resource-type>choice group</resource-type>
         <resource-name>Campaign,Slice,Role,Offer</resource-name>
         <actions>create,read,update,delete,own</actions>
      </permission>
   </permissions>
</application-role>
```

Example Case 2:

To create an application role that does not have CRUD permission on Slice and Role, you would enter the following code in `security.xml`. This code indicates that, except for Role and Slice choices, all other choices have permission for CRUD operations.

```
<application-role>
   <name>CLMSliceManager</name>
   <permissions>
      <permission>
         <resource-type>choice group</resource-type>
         <resource-name>^Role,Slice</resource-name>
         <actions>create,read,update,delete,own</actions>
      </permission>
   </permissions>
</application-role>
```

To create an application role having permission for both Roles and Slices perspectives:

1. Use `Roles` and `Slices` as the value for the `<resource-name>` attribute for the perspective type.
2. Use `_all_` as the value for the `<resource-name>` attribute of perspective type.

Example Case 1:

This example gives the user view permission on the Roles, Slices, and Campaign perspective.

```
<permission>
  <resource-type>perspective</resource-type>
  <resource-name>Roles,Slices,Campaigns</resource-name>
  <actions>view</actions>
</permission>
```

Example Case 2:

This example gives the user view permission on all perspectives except Roles and Slices.

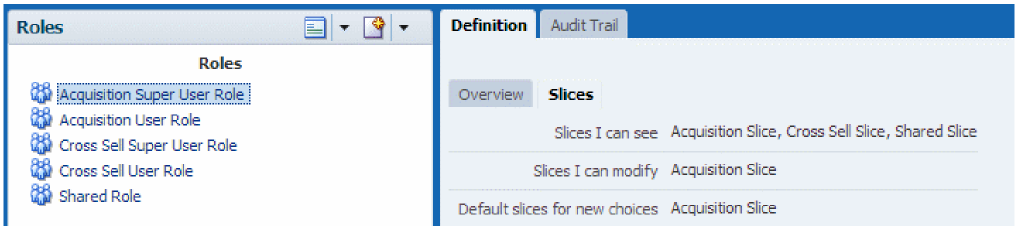
```
<permission>
  <resource-type>perspective</resource-type>
  <resource-name>^Roles,Slices </resource-name>
  <actions>view</actions>
</permission>
```

> **Note:** If `security.xml` or any of its related files, such as `groups.xml`, is modified or added, then build, generate, and deploy the application by following the steps in Section 2.3.5.4, "Generate the Decision Manager Application".

2.3.5.3 Create an Application Role for a Slice-based Role in security.xml

To add or create an application role in Oracle RTD Decision Manager metadata and to map it to Decision Manager Slice based Role choices (in Decision Manager metadata database), do the following:

The following illustration lists the Slice-based Roles defined in CLM database.



Roles

- Acquisition Super User Role
- Acquisition User Role
- Cross Sell Super User Role
- Cross Sell User Role
- Shared Role

Slices I can see: Acquisition Slice, Cross Sell Slice, Shared Slice

Slices I can modify: Acquisition Slice

Default slices for new choices: Acquisition Slice

1. Note the names of the Roles that you want to make available to users.
2. In `clm/Build/metadata/ref-slices`, edit `security.xml`, or create a different file, such as `groups.xml`.
3. Create an entry in `security.xml` or `groups.xml` as shown here:

```
<application-role>
  <name>Shared Role</name>
</application-role>
<application-role>
  <name>Acquisition Super User Role</name>
</application-role>
<application-role>
  <name>Acquisition User Role</name>
</application-role>
```

Note that the names of the application roles should match those for the Role based choices created in the Decision Manager database.

The Shared Role choice created in the Decision Manager database should have an entry for the application role in `security.xml` where the `<name>` attribute has `Shared Role` as its value, as shown in the following example.

```
<application-role>
  <name>Shared Role</name>
</application-role>
```

Note: If security.xml or any of its related files, like groups.xml is modified or added, then build, generate, and deploy the application by following the steps in Section 2.3.5.4, "Generate the Decision Manager Application".

2.3.5.4 Generate the Decision Manager Application

To generate the Decision Manager application, do the following:

1. Open a command prompt and navigate to the clm/Build directory.
2. Run the following command:

   ```
   ant generate -Dmetadata.module=ref-slices
   ```

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

2.3.5.5 Deploy RTD Decision Manager

Deploy the Decision Manager application to WebLogic Server. For deployment instructions, see Section 1.2.3.5, "Oracle RTD Decision Management Application Deployment".

2.3.6 Mapping Users or Groups to Slice-based Roles

This section explains how to map the users and groups created in your organization to the Slice-based Decision Manager application roles and/or the Standard roles.

Before you begin, ensure that Slice-based Role choices are already created in Decision Manager following instruction given in user guide.

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

To map users or groups to Decision Manager application roles, do the following:

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 RTD Base Marketing (with slices) deployment: Application Deployments then Internal Applications then RTD Base Marketing (with slices).
4. In the RTD Base Marketing (with slices) window, from the Application Deployment dropdown menu, select Security then Application Roles and click Search application roles.
5. Click CLMSliceAdministrator, CLMSliceAuthor or CLMSliceConsumer depending on the level of permissions you want for your users or groups.
6. Click Edit the selected application role.
7. Click **Add roles**.
8. Repeat steps 4 through step 7 to add Slice based Role like Shared Role, Cross Sell Super User Role, and so on.
9. In the Type dropdown list, change Application role to User.
10. Search for your user and click **OK**.

11. To add a group, click **Add roles** and in the Type drop-down list, change Application role to Group.
12. Search for your group and click **OK**.
13. Click **OK**.

2.4 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