Oracle® Real-Time Decisions

Installation and Administration Guide

Release 3.2

E52406-02

June 2014

Describes how to install, configure, and administer Oracle Real-Time Decisions (Oracle RTD), how to set up authentication and authorization for maintaining security, and how to manage Oracle RTD MBeans.

Oracle Real-Time Decisions Installation and Administration Guide, Release 3.2

E52406-02

Primary Author: Oracle Corporation

Contributors: Oracle Business Intelligence development, product management, and quality assurance teams

Contents

5 Configuring Security for Oracle Real-Time Decisions

10 Performance Monitoring

11 Managing Oracle Real-Time Decisions

A System Log and Configuration Files

B Exceptions

Preface

Oracle Real-Time Decisions (Oracle RTD) enables you to develop adaptive enterprise software solutions. These adaptive solutions continuously learn from business process transactions while they execute and optimize each transaction, in real time, by way of rules and predictive models.

This document provides information about installing and administering Oracle RTD Release 3.2. It explains how to install and configure Oracle RTD, set up authentication for maintaining security, and manage Oracle RTD MBeans.

Audience

This document is intended for administrators of Oracle RTD. Oracle RTD administrators should have a working knowledge of how to install and administer enterprise-level applications, and should know how to use a JMX console.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle RTD Release 3.2 documentation set:

- *Oracle Real-Time Decisions Platform Developer's Guide*
- *Oracle Real-Time Decisions Decision Center User's Guide*
- *Oracle Real-Time Decisions Release Notes*

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.

A Note About Path Names

In this document, forward slashes (/) are used in path names for steps that can be performed on multiple platforms. If you are typing the path on a UNIX system, you can enter the text as it is given. If you are typing the path on a Windows system, you must change each forward slash (/) to a back slash (\).

In this document, back slashes (\) are used in path names that appear in steps or examples that are specific to the Windows operating system.

General Note: When you cut and paste multi-line code or command text, extra carriage return characters get added to the output, and will cause errors if they are not removed.

1

Preparing for Installation

Oracle RTD is supported on both Windows and UNIX platforms, and runs on Oracle WebLogic and IBM WebSphere application servers. For detailed information about supported platforms, refer to *Oracle Real-Time Decisions System Requirements and Supported Platforms* for Version 3.2.

This chapter contains the following topics:

- Section 1.1, "About Oracle Real-Time Decisions Installation"
- Section 1.2, "Installing the Database Server,"
- Section 1.3, "Installing the Application Server"
- Section 1.4, "Additional Java Development Kit (JDK) Tasks"
- Section 1.5, "Additional Preinstallation Tasks"

1.1 About Oracle Real-Time Decisions Installation

Oracle RTD is a real-time decision engine software platform that consists of the following components:

- Real-Time Decision Server
- Decision Studio
- Decision Center
- Load Generator
- Administration (JMX)

Real-Time Decision Server can be installed on either Windows or UNIX platforms. Decision Studio and Load Generator are client components that can only run on Windows.

Real-Time Decision Server requires its own database, called the Oracle RTD Database, to store objects and metadata for inline services created in Decision Studio.

To install Oracle RTD, you need to:

1. Install the database server, and if required by your chosen application server, a Java Development Kit.
2. Install files for the Real-Time Decision Server and Oracle RTD clients, and create and initialize the database.
3. Install the application server of your choice.
4. Configure the data source for the Oracle RTD Database.

5. Deploy and configure Real-Time Decision Server using the administration console for your application server, and other tools.
6. Configure authentication settings.
7. Set up access to enterprise data sources.

1.2 Installing the Database Server

Install the database server you want to use as the Oracle RTD Database. This database must be one of the following: Oracle Database or Microsoft SQL Server. For information about database installation, refer to your database documentation. For detailed information about supported database versions, refer to *Oracle Real-Time Decisions System Requirements and Supported Platforms* for Version 3.2, available from the Oracle RTD Documentation Web site at:

`http://www.oracle.com/technetwork/middleware/real-time-decisions/documentation/index.html`

Note the following:

- If you are using SQL Server, you may install your database server either as the default (unnamed) instance or as a named instance.
- If you are using Oracle Database, you must install the user_lock package. Refer to the Oracle Database documentation for information about installing optional Oracle packages.
- You can install your database server on the same computer where you plan to install Oracle RTD, or on a different computer.

1.3 Installing the Application Server

Real-Time Decision Server runs on one of the following application servers: WebSphere or WebLogic. For detailed information about supported application server versions, refer to *Oracle Real-Time Decisions System Requirements and Supported Platforms* for Version 3.2, available from the Oracle RTD Documentation Web site at:

`http://www.oracle.com/technetwork/middleware/real-time-decisions/For instdocumentation/index.html`

Note: For up to date information on specific versions of your application server, read all references to that application server in *Oracle Real-Time Decisions Release Notes*.

For WebLogic systems, see Chapter 3, "Installing and Configuring WebLogic for Oracle Real-Time Decisions."

For WebSphere systems, see Chapter 4, "Installing and Configuring WebSphere for Oracle Real-Time Decisions."

1.4 Additional Java Development Kit (JDK) Tasks

WebLogic and WebSphere install their own JDK.

For client tools (Decision Studio, Load Generator, SDDBTool and JConsole), you must install a 32-bit JDK. Create a system environment variable called JAVA_HOME and set its value to the full path name of the install location of the JDK.

Modify the system environment variable PATH by adding `%JAVA_HOME%\bin;` to the beginning of the existing value.

For example, if the existing PATH value is `abc;'`, then the new value should be:

```
'%JAVA_HOME%\bin;abc;'
```

1.5 Additional Preinstallation Tasks

Close all open applications before you install Oracle RTD. In particular, shut down any applications or processes that use port 8080, the default Oracle RTD port.

In addition, shut down any real-time indexing software, such as Google Desktop, before installing Oracle RTD. Running real-time indexing software on the computer where Real-Time Decision Server is running can adversely impact Oracle RTD performance.

2

Installing the Real-Time Decision Server

This chapter describes the steps involved in the installation of Oracle RTD, the creation of the Oracle RTD database, and the (optional) population of database tables that provide the data source referenced by the Inline Services released with Oracle RTD.

The Real-Time Decision Server must be installed on the computer where your chosen application server is resident.

This chapter contains the following topics:

- Section 2.1, "Installing Oracle Real-Time Decisions Files"
- Section 2.2, "Creating and Initializing the Oracle RTD Database"
- Section 2.3, "Populating the CrossSell Example Data (Optional)"
- Section 2.4, "Populating the DC_Demo Example Data (Optional)"
- Section 2.5, "Using SSL with Oracle Real-Time Decisions"

2.1 Installing Oracle Real-Time Decisions Files

Real-Time Decision Server can run on either Windows or UNIX. The Oracle RTD client tools, such as Decision Studio, Load Generator, and SDDBTool, can only run on Windows.

This section contains the following topics:

- Section 2.1.1, "Installing Oracle RTD on a Single Windows Computer"
- Section 2.1.2, "Installing Real-Time Decision Server on UNIX"
- Section 2.1.3, "About the Oracle RTD Run-Time Environment"

2.1.1 Installing Oracle RTD on a Single Windows Computer

To install all Oracle RTD components on a single Windows computer, follow these steps:

1. Unzip the appropriate Oracle RTD archive file into the directory where you want to install the Oracle RTD files. Be sure to choose the correct archive file for your operating system and application server type. For example, the file format for Windows and WebLogic is rtd-weblogic-3.2.0.0.0.<*build_number*>.zip.

   After you extract the files, a directory called `OracleBI\RTD` is created that contains all the Oracle RTD files. This directory is your *`RTD_HOME`*.

2. If you are using WebLogic, the `.ear` file for the Oracle RTD application must be expanded before it can be deployed on the WebLogic application server. To do this, follow these steps:
   a. Go to *`RTD_HOME`*`\package` and create a directory called `expanded` to hold the contents of the `RTD.ear` file.
   b. Use WinZip to extract the *`RTD_HOME`*`\package\RTD.ear` file into the *`RTD_HOME`*`\package\expanded` directory.

2.1.2 Installing Real-Time Decision Server on UNIX

To install Real-Time Decision Server on a UNIX computer, and install the Oracle RTD client tools on a Windows computer, follow these steps:

1. To install Real-Time Decision Server on UNIX, use `cpio` to extract the appropriate archive file into the directory on the UNIX computer where you want to install the Real-Time Decision Server files. Be sure to choose the correct archive file for your operating system and application server type. For example, the file format for Windows and WebLogic is rtd-weblogic-3.2.0.0.0.<*build_number*>.cpio.

   After you extract the files, a directory called `OracleBI/RTD` is created that contains all the Real-Time Decision Server files. This directory is your *`RTD_HOME`* on the UNIX computer.
2. If you are using WebLogic, the `.ear` file for the Oracle RTD application must be expanded before it can be deployed on the WebLogic application server. To do this, follow these steps:
   a. Go to *`RTD_HOME`*`/package` and create a directory called `expanded` to hold the contents of the `RTD.ear` file. Note that the `expanded` folder should not contain the actual `RTD.ear` file, just its contents.
   b. Use the `jar` utility to extract the *`RTD_HOME`*`/package/RTD.ear` file into the *`RTD_HOME`*`/package/expanded` directory.
3. To install the Oracle RTD client tools on Windows, unzip the appropriate Oracle RTD archive file into the directory where you want to install the Oracle RTD client tools. Be sure to choose the correct archive file for your operating system and application server type.

   After you extract the files, a directory called `OracleBI\RTD` is created that contains all the Oracle RTD files. This directory is your *`RTD_HOME`* on the Windows computer.

2.1.3 About the Oracle RTD Run-Time Environment

In this document, *`RTD_HOME`* refers to the directory where the Oracle RTD files were installed. Because a directory structure is created when you extract the Oracle RTD files, the last part of the path will always be `OracleBI/RTD`. For example, if you extracted the Oracle RTD files to `C:\` on Windows, the *`RTD_HOME`* is `C:\OracleBI\RTD`.

Some Oracle RTD management tasks, such as viewing Oracle RTD logs, require you to locate the *`RTD_RUNTIME_HOME`*. The location of *`RTD_RUNTIME_HOME`* depends on the application server you are using:

- If you are using WebSphere, *`RTD_RUNTIME_HOME`* is *`WEBSPHERE_HOME`*`/AppServer/profiles/`*`profile_name`*`/installedApps/`*`cell_name`*`/OracleRTD.ear`.

- If you are using WebLogic, *`RTD_RUNTIME_HOME`* is the directory where the expanded `RTD.ear` folder is deployed. For example, *`MW_HOME`*`/user_projects/domains/`*`domain_name`*`/servers/`*`server_name`*`/stage/OracleRTD/`.

2.2 Creating and Initializing the Oracle RTD Database

After you have installed the Oracle RTD files, you must create and initialize the Oracle RTD Database.

Oracle RTD requires two database users: an administrative user, and a run-time user. The administrative user needs to be able to create tables and stored procedures, and is only used when the database is being initialized during initial Oracle RTD configuration. The administrative user is never used during Oracle RTD run time. The run-time database user is used by Oracle RTD during run time to access information in the database.

There are two ways to initialize the Oracle RTD Database. You can either run SDDBTool, an Oracle RTD Windows client application, or you can run command line scripts provided by Oracle RTD.

This section contains the following topics:

- Section 2.2.1, "Creating the Oracle RTD Database"
- Section 2.2.2, "Initializing the Oracle RTD Database Using SDDBTool"
- Section 2.2.3, "Initializing the Oracle RTD Database Using Command Line Scripts"

2.2.1 Creating the Oracle RTD Database

For your Oracle RTD Database, you must provide the net service name and a valid user name and password. Refer to your Oracle Database documentation for information about how to set up net services.

On your Oracle RTD database, as the database type is Oracle, the administration user and the run-time user must be the same user. This user needs to have the database roles `Connect` and `Resource`. The `Resource` role can be revoked after the database has been initialized. If you do revoke the `Resource` role, set the quota for the tablespace of the run-time user to an appropriate level, or to `unlimited`, depending on your database policies.

You must install Oracle Administrator Client on the computer where you plan to run SDDBTool. See Section 2.2.2, "Initializing the Oracle RTD Database Using SDDBTool" for more information about SDDBTool.

2.2.2 Initializing the Oracle RTD Database Using SDDBTool

You can initialize the Oracle RTD Database using SDDBTool, an Oracle RTD Windows client application. Before using SDDBTool to initialize the database, perform the following steps:

- Ensure that Real-Time Decision Server is not running
- Perform the instructions in *`RTD_HOME`*`\lib\jdbc\readme.txt` to install `ojdbc6.jar` into *`RTD_HOME`*`\lib\jdbc\`.

To initialize the Oracle RTD Database using SDDBTool:

1. Ensure that your database server is running.

2. Edit the file *RTD_HOME*`\scripts\sql\Oracle\SDTablespaceMap.txt` on the computer where you installed the Oracle RTD client tools. This file allows you to control the allocation of Real-Time Decision Server tables to your Oracle tablespaces. Read the instructions in the file for more information. After you edit the file, save and close the file.
3. Open SDDBTool by running *RTD_HOME*`\scripts\SDDBTool.cmd` on the computer where you installed the Oracle RTD client tools. Then, choose your database type.
4. Enter your database settings, as follows:
   - Database Host
   - Database Port
   - Database Name
   - Database Runtime User
   - Database Administrative User
   - Database Administrative User Password

   Note: The database administrative user you provide must have rights to create tables and stored procedures on the database. The run-time user is used to access system data at run time.

   As the database type is Oracle, the Database Runtime User and the Database Administrative User are the same user.

   ---

5. Click **Next**.
6. If your database is a RAC database, select the "RAC database" option.

   To enable throughput improvements on Learning Server processing, and if your database supports partitioning, select the “Enable partitioning” option.

   Select **Initialize** or **Upgrade**. Initialize creates the data and data source needed to run Oracle RTD. Upgrade will upgrade from a previous version to the current version.

If you do not plan to use the example CrossSell Inline Service, and you are using Oracle Database for your Oracle RTD Database, you can revoke the `Resource` role from the database user after you complete database initialization. See Section 2.3, "Populating the CrossSell Example Data (Optional)" for more information about the example CrossSell Inline Service.

2.2.3 Initializing the Oracle RTD Database Using Command Line Scripts

You can initialize the Oracle RTD Database using command line scripts. Before using the scripts to initialize the database, make sure that Real-Time Decision Server is not running.

Under the directory *RTD_HOME*`\scripts\sql`, there is a directory, **OracleStatic**.

This directory contains a `readme.txt` file, which contains instructions on how to use the other files provided in the directory.

Follow the instructions in the `readme.txt` file.

Note: The OracleStatic directory does not contain an `initialize.sql` file. The OracleStatic `readme.txt` file contains instructions on how to generate the sql to initialize the database from the `initalize.template` file.

If you do not plan to use the example CrossSell Inline Service, and you are using Oracle Database for your Oracle RTD Database, you can revoke the `Resource` role from the database user after you complete database initialization. See Section 2.3, "Populating the CrossSell Example Data (Optional)" for more information about the example CrossSell Inline Service.

2.3 Populating the CrossSell Example Data (Optional)

An example Inline Service, called CrossSell, is included with Oracle RTD. To use this sample inline service, you must create and populate three tables, `CrossSellCustomers`, `CrossSellResponses`, and `CrossSellBestOffer` in the Oracle RTD Database. To do this, run the script `InitAppDB` on the Windows computer where you installed the Oracle RTD client tools.

`InitAppDB` is located with the example Inline Service. Using a command prompt, run the following script:

- *`RTD_HOME`*`\examples\CrossSell\etc\data\Oracle\initappdb.cmd`

This script takes the following parameters:

```
InitAppDB RTD_HOME db_host db_port db_name db_runtime_user db_admin_user db_admin_
password
```

Table 2–1 describes the parameters for the `InitAppDB` script.

Table 2–1 Parameters for InitAppDB Script

Parameter	Description
`RTD_HOME`	The full path of the directory where the Oracle RTD files are installed.
`db_host`	The name of the computer hosting the database server.
`db_port`	The database port number.
`db_name`	The database name.
`db_runtime_user`[1]	The user name of the run-time user for the system.
`db_admin_user`	The name of a user that has rights to create tables and stored procedures on the database.
`db_admin_password`	The password of the administrative user.

[1] For Oracle Database, the *`db_runtime_user`* and *`db_admin_user`* are the same user.

For your Oracle RTD Database, you can revoke the `Resource` role from the database user after you run the `InitAppDB` script.

2.4 Populating the DC_Demo Example Data (Optional)

Another example Inline Service, called DC_Demo, is included with Oracle RTD, to demonstrate dynamic choices and external rules. To use this sample inline service, you must first create and populate a sample database table WebOffers.

To do this, run the script `InitAppDB` on the Windows computer where you installed the Oracle RTD client tools.

`InitAppDB` is located with the example Inline Service. Using a command prompt, run the following script:

- *RTD_HOME*`\examples\DC_Demo\etc\data\Oracle\initappdb.cmd`

This script takes the following parameters:

```
InitAppDB RTD_HOME db_host db_port db_name db_runtime_user db_admin_user db_admin_
password
```

See Table 2–1 for the parameters for the `InitAppDB` script.

2.5 Using SSL with Oracle Real-Time Decisions

If you want to use SSL for client connections to Oracle RTD (recommended), and if you want to use the Oracle RTD keystore and truststore, you must first change the passwords for the keystore and truststore. To do this, follow these steps:

1. Run the following keytool command to change the keystore password:

   ```
   JAVA_HOME/bin/keytool -storepasswd -new new_password -keystore RTD_
   HOME/etc/ssl/sdserver.keystore
   ```

   Make sure to replace *RTD_HOME* with the actual Oracle RTD installation path. For *new_password*, provide the new password you want to use for the Oracle RTD keystore.

   When the script prompts for the existing keystore password, enter `tc-ssl`.

   Note: If you cut and paste the keytool command syntax from this document, the hyphen character (-) may be copied as a long hyphen rather than a short hyphen. If you receive an error, try re-typing the hyphen characters in the command.

2. Run the following keytool command to change the truststore password:

   ```
   JAVA_HOME/bin/keytool -storepasswd -new new_password -keystore RTD_
   HOME/etc/ssl/sdtrust.store
   ```

   Make sure to replace *RTD_HOME* with the actual Oracle RTD installation path. For *new_password*, provide the new password you want to use for the Oracle RTD truststore.

   When the script prompts for the existing truststore password, enter `tc-ssl`.

3. Run the following keytool command to change the server private key password:

   ```
   JAVA_HOME/bin/keytool -keypasswd -alias tc-ssl -keypass tc-ssl -new new_
   password -keystore RTD_HOME/etc/ssl/sdserver.keystore
   ```

   Make sure to replace *RTD_HOME* with the actual Oracle RTD installation path. For *new_password*, provide the password you want to use for the server private key.

When the script prompts for the keystore password, enter the password you provided for the keystore in Step 1.

2.5.1 Additional Information for SSL-Only Configurations

For greater security, when setting up SSL, you should disable the regular HTTP port to ensure that all client connections are routed through the SSL port.

> **Note:** Before you disable the regular HTTP port, make sure that SSL is working and that you can connect to Oracle RTD using the HTTPS protocol, as in https://*server*:*ssl_port*/ui/.

When Oracle RTD is deployed on an application server, you may need to perform the following additional steps when setting up an SSL-only configuration:

1. Run the following keytool command to import the Oracle RTD certificate into the cacerts for your application server:

   ```
   keytool -import -file RTD_HOME/etc/ssl/sdserver.cer -keystore cacerts
   ```

   The file `keytool.exe` is located in the standard `jdk/bin` directory for the JDK used by the application server.

2. When prompted, enter the keystore password for the application server's `cacerts` file.

   The default location of cacerts is in the standard `jdk/jre/lib/security` directory for the JDK used by the application server.

3. When prompted to trust the certificate, enter **yes**.

This procedure is useful for test environments. For production systems, you should use your own certificate rather than the test certificate that ships with Oracle RTD.

3

Installing and Configuring WebLogic for Oracle Real-Time Decisions

This chapter describes how to install and configure the Real-Time Decision Server on WebLogic.

> **Note:** Although the Real-Time Decision Server runs on either UNIX or Windows, the Oracle RTD client tools must be run from a Windows platform.

There are many system topologies possible for Oracle RTD systems set up under WebLogic. The complete setup of each system depends on a variety of factors, including the complexity and requirements of the system topology and the business implementation strategy.

In this section, it is assumed that you perform a typical system setup in two broad stages:

1. A basic setup stage, where you install and configure Oracle RTD into a simple WebLogic domain, on the computer where you want to run Real-Time Decision Server.

   At the end of this stage, you can then perform a series of basic verification tasks, to test if your system installation and configuration has succeeded.

2. A "post-installation" stage, where you select extra, optional facilities and features to add to and enhance the basic system.

   At the end of this stage, you can optionally perform one or more of the basic verification tasks. These do not test the new features, but may establish if any new errors have arisen.

The setup tasks in the "post-installation" stage are largely standalone and independent of each other, therefore you can perform them in any order. For the basic setup stage, Oracle advises that you follow the sequence of the tasks and the steps within the tasks as described in this section.

This chapter includes the following topics:

- Section 3.1, "Basic Setup Tasks"
- Section 3.2, "Verification Tasks"
- Section 3.3, "Post-Installation Tasks"
- Section 3.4, "Uninstalling the Oracle Real-Time Decisions Application from WebLogic"

- Section 3.5, "Configuring SSL for Real-Time Decision Server (Recommended)"

3.1 Basic Setup Tasks

The basic tasks for setting up WebLogic and configuring it for Oracle RTD appear in the following sections *(for convenience the sections are grouped into the broad Install, Configure and Deploy categories)*:

Install WebLogic + Create Domain

- Section 3.1.1, "Installing WebLogic"
- Section 3.1.2, "Creating a WebLogic Domain"

Configure WebLogic for Oracle RTD

- Section 3.1.3, "Configuring JVM Properties for Managed Server"
- Section 3.1.4, "Configuring JDBC Source for Oracle RTD Database"
- Section 3.1.5, "Configuring Oracle RTD Security in WebLogic"

Deploy Oracle RTD in WebLogic

- Section 3.1.6, "Deploying Oracle RTD to WebLogic"
- Section 3.1.7, "Restarting WebLogic Server"
- Section 3.1.8, "Starting Clustered Servers"
- Section 3.1.9, "Starting OracleRTD"

3.1.1 Installing WebLogic

For general information about installing WebLogic, see *Oracle Fusion Middleware Installation Guide for Oracle WebLogic Server*. Unless otherwise stated, the link references in this section are to topics in that manual.

There are several modes of WebLogic installation - Graphical, Console, Silent. You can use whichever mode is convenient - the following description assumes the Graphical installation mode.

To install WebLogic, follow the instructions in Section 3.1, *Starting the Installation Program in Graphical Mode*.

The screens that are the framework for the graphical mode installation are described in detail in the following sections:

- Section 3.2 *Installation Screens*
- Appendix A *Oracle WebLogic Server Installation Screens*

Table 3–1 describes the main WebLogic installation screens and the options for a basic WebLogic installation (the "Screen" column shows the screen topic sections as described in Appendix A *Oracle WebLogic Server Installation Screens*).

Table 3–1 WebLogic Installation Screens

Screen	Screen Title	Notes
A.1	Welcome	
A.2	Choose Middleware Home Directory	Create a new Middleware Home For example, C:\Oracle\Middleware
A.3	Register for Security Updates	N/A
A.4	Choose Install Type	Typical - for WebLogic Server + Coherence *Note: if you want to switch off Coherence, select Custom.*
A.5	Choose Products and Components	If you selected Custom on A.4, you can uncheck Coherence.
A.6	JDK Selection	If bundled JDK available, select it. Otherwise, select the local JDK you want to use
A.7	Choose Product Installation Directories	For example, C:\Oracle\Middleware\wlserver_10.3
A.8	Confirm Product Installation Directory	N/A
A.9	Install Windows Service	This is to enable Node Manager to be installed as a Windows service. Select Yes.
A.10	Choose Shortcut Location	Leave the default: "All Users" Start Menu
A.11	Installation Summary	N/A
A.12	Status	N/A
A.13	Installation Complete	**At end, launch Quickstart**. This launches the Configuration Wizard, where you can create a WebLogic domain

3.1.2 Creating a WebLogic Domain

For conceptual information about WebLogic domains, see *Oracle Fusion Middleware Understanding Domain Configuration for Oracle WebLogic Server*, and in particular, Section 2 *Understanding Oracle WebLogic Server Domains*.

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

- Section 2.3 *Starting the Configuration Wizard in Graphical Mode*
- Section 2.4 *Creating a WebLogic Domain in Graphical Mode*
- Section 4 *Configuration Wizard Screens*

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

- 1 Admin Server
- 1 Machine
- 1 Cluster (with 1 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 Section 2.3 *Starting the Configuration Wizard in Graphical Mode*, or through QuickStart (select Getting Started with WebLogic Server 10.3.6).

Then follow the instructions as described in Section 2.4 *Creating a WebLogic Domain in Graphical Mode*.

Table 3–2 describes the main Configuration Wizard screens and the options to create a simple, single-cluster WebLogic domain (the "Screen" column shows the screen numbers as specified in Section 2.4 *Creating a WebLogic Domain in Graphical Mode*).

In the table, the suggested example names for the elements created are AS1 for the Admin Server, MC1 for the Machine, C1 for the Cluster, and MS1 for the Managed Server.

Table 3–2 WebLogic Configuration Wizard Screens for Creating a Simple Domain

Screen	Screen Title	Notes
1	Welcome	Create new domain
2	Select Domain Source	Generate a domain configured automatically to support the following products
3	Specify Domain Name and Location	Domain name (suggestion): RTD_domain (Goes under MW_HOME\user_projects\domains)
4	Configure Administrator User Name and Password	Used for all WebLogic administrator processes (required for Production Mode startup)
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
14	Select Optional Configuration	Select: - Administration Server - Managed Servers, Clusters and Machines
15	Configure Administration Server	Example: Administration Server (AS1) listens on 7001.
17	Configure Managed Servers	Example: Managed Server (MS1) listens on 7003.
18	Configure Clusters	Example: Cluster Name C1
19	Assign Servers to Clusters	Assign MS1 to C1
21	Configure Machines	Example: Machine Name MC1
22	Assign Servers to Machines	Assign AS1 and MS1 to MC1
27	Configuration Summary	
28	Creating Domain	**At end, start Admin Server**

3.1.3 Configuring JVM Properties for Managed Server

To add JVM and startup arguments to the managed server, perform the following steps in the WebLogic Administration Console:

> **Note:** *See General Note if you intend to cut and paste multi-line code or command text from this section.*

1. Navigate the path:

   Environment > Servers > *managed_server_name* > Configuration > Server Start tab.

2. In the Arguments field, you must enter a string of parameters, all on one line.

   Use the following string as a template:

   ```
   -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.S
   DEMFRegistry -Drtd.instanceName=<managed_server_name>
   -Drtd.HTTPHostAndPort=<host>:<port> -Dcom.sun.management.jmxremote=true
   -Dcom.sun.management.jmxremote.port=<JConsole_port>
   -Dcom.sun.management.jmxremote.authenticate=false
   -Dcom.sun.management.jmxremote.ssl=false
   -Dweblogic.wsee.skip.async.response=true -XX:MaxPermSize=1024m
   ```

   Notes:

   1. For Sun JVM, leave `-XX:MaxPermSize` as `1024m`; for JRocket, you can remove this parameter.
   2. The system property, **rtd.skipIlsTestLoad**, can be used to bypass test loading of an Inline Service (ILS) during ILS deployment. If unspecified in the JVM start parameters, the property has a default value of false, which means ILS test loading will be performed during ILS deployment.

      If the logging level is set to DEBUG, a message similar to the following can be seen in the server log file:

      ```
      2013-12-10 11:04:10,142 DEBUG [DeployAppCommand] Performing
      test load on CrossSell with deployment state 5.
      ```

      Set the property to true, that is, add **-Drtd.skipIlsTestLoad=true** to bypass ILS test loading. This may be helpful in ILS redeployments from Decision Center in a clustered environment.
   3. If your Inline Services have many external rules, add the following JVM parameter to improve the initial load time of these external rules:

      -Dsun.reflect.inflationThreshold=2147483647

   For the three highlighted parameters, specify the values that are specific to your system:

   - **`rtd.instanceName`** identifies the managed server in the cluster

     Note: The instanceName value can only contain alphanumerics, underscore, dash and period.

   - **`rtd.HTTPHostAndPort`** for communication between managed servers such a during request forwarding
   - **`com.sun.management.jmxremote.port`** is used when launching JConsole

   For example *(no new-line or carriage-return characters allowed)*:

   ```
   -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.S
   DEMFRegistry -Drtd.instanceName=MS1 -Drtd.HTTPHostAndPort=srvmach01:7003
   -Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.port=12302
   -Dcom.sun.management.jmxremote.authenticate=false
   -Dcom.sun.management.jmxremote.ssl=false
   -Dweblogic.wsee.skip.async.response=true -XX:MaxPermSize=1024m
   ```

3. Save.

3.1.4 Configuring JDBC Source for Oracle RTD Database

For general information about creating JDBC Sources, see *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*, and in particular, 3.3 *Creating a JDBC Data Source*.

To create the JDBC data source for the Oracle RTD database, perform the following steps:

1. In the tree to the left under Change Center, click **Lock & Edit.**
2. Navigate the path **Services** -> **Data Sources** -> **New** -> **Generic Data Source**.
3. Click **New**.
4. For **Name**, enter the name for the data source. For example, `RTD_DS`.
5. For **JNDI Name**, enter `SDDS` *(this must be the exact name)*.
6. For **Database Type**, select **Oracle**.
7. For **Database Driver**, select **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later**.
8. Click **Next**.
9. Uncheck the **Supports Global Transactions** checkbox.
10. Click **Next**.
11. On the Connection Properties page, perform these steps:
    a. For **Database Name**, enter the name of the Oracle RTD Database you created in Section 2.2, "Creating and Initializing the Oracle RTD Database."
    b. For **Host Name**, enter the name of the computer hosting the database server.
    c. For **Port**, enter the port number on the database server used to connect to the database.
    d. For **Database User Name**, enter the name of the database run-time user.
    e. For **Password**, enter the password of the database run-time user, then click **Next**.
12. On the Test Database Connection page, leave all the settings already filled, except enter `SDAPPS` for **Test Table Name**.
13. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**.
14. Under Clusters, select the checkbox for your cluster.
15. Click **Finish**.

    (Optional) If you want to improve performance, you should consider turning off the "Wrap Data Types" option, by performing these steps:

    a. Select the data source that you just created.
    b. Under the Configuration tab, click on the Connection Pool sub tab.
    c. Scroll down and expand the Advanced section.
    d. Deselect **Wrap Data Types**.
    e. Click **Save**.
16. In the tree to the left under Change Center, click **Activate Changes**.

3.1.5 Configuring Oracle RTD Security in WebLogic

Note: For general information about roles and permissions in Oracle RTD, see Chapter 5, "Configuring Security for Oracle Real-Time Decisions."

In WebLogic, Oracle RTD roles are defined in terms of user groups. A user is in a role if the user is in any of the groups referenced by the role. So the process is to create the groups, then create the roles that reference the groups. Users may be assigned to or removed from the groups at any time to add them or remove them from the referencing roles.

The following topics describe the stages of the process of creating Oracle RTD roles for WebLogic:

- Section 3.1.5.1, "Creating User Groups for Standard Roles"
- Section 3.1.5.2, "Creating User Groups for Custom Roles (Optional)"
- Section 3.1.5.3, "Specifying Subgroups"
- Section 3.1.5.4, "Creating Roles"
- Section 3.1.5.5, "Creating the Oracle RTD Administrator and Other Users"

3.1.5.1 Creating User Groups for Standard Roles

To create user groups for standard roles, perform the following steps:

1. Open the WebLogic Administration Console.
2. In the tree on the left, select Security Realms.
3. In the window on the right, select the security realm to use. The default realm is myrealm.
4. Select the Users and Groups tab.
5. Select the Groups sub-tab.
6. Create each of these groups, or names of your own choosing:
    - RTDUserGroup
    - RTDAdminGroup
    - RTDBatchAdminGroup
    - RTDChoiceEditorGroup
    - RTDDCEditorGroup
    - RTDDCUserGroup
    - RTDStudioDeployerGroup
    - RTDStudioDownloaderGroup

3.1.5.2 Creating User Groups for Custom Roles (Optional)

Create enterprise-specific groups, if Inline Service development and access is to be controlled for separate Inline Services.

For example:

- ILS2DevGroup - Developers for Inline Service ILS2

- ILS2UserGroup - Business users for Inline Service ILS2

3.1.5.3 Specifying Subgroups

Select each of the standard and custom Oracle RTD groups in turn, except for RTDUserGroup, and make each of them a member of the group RTDUserGroup. This will allow any user to be automatically added to the RTDUserGroup when they are added to one of the other Oracle RTD groups.

The individual steps for each group are as follows:

1. Select the group.
2. Select its Membership tab.
3. Select RTDUserGroup in the Available column and move it to the Chosen column.
4. Click Save.

3.1.5.4 Creating Roles

Create standard Oracle RTD roles and any custom roles needed by the enterprise. Standard roles are the ones to which Oracle RTD automatically assigns realm permissions. Custom roles require cluster permission assignments through JMX.

The following table shows the standard roles and some custom roles, as well as the groups to associate with the roles, to illustrate the steps described in this section.

Table 3–3 Role and Group Associations

Role	Group	Standard or Custom
RTDUsers	RTDUserGroup	Standard
RTDAdministrators	RTDAdminGroup	Standard
RTDDecisionCenterEditors	RTDDCEditorGroup	Standard
RTDDecisionCenterUsers	RTDDCUserGroup	Standard
RTDStudioDeployers	RTDStudioDeployerGroup	Standard
RTDStudioDownloaders	RTDStudioDownloaderGroup	Standard
RTDBatchAdministrators	RTDBatchAdminGroup	Standard
RTDChoiceEditors	RTDChoiceEditorGroup	Standard
ILS2Developers	ILS2DevGroup	Custom - an example
ILS2Users	ILS2UserGroup	Custom - an example

Perform the following steps:

1. Select the Roles and Policies tab of the security realm.
2. Expand the Global Roles item in the Name column of the Roles.
3. Select Roles.

   The following steps enable you to create the roles and to reference the corresponding group, as shown in the preceding table.
4. For each role, perform the following steps:
   1. Click New to create a new global role.
   2. Enter the role name, as in Table 3–3.

3. Click OK.
4. Open the new role by clicking on its name in the global roles list.
5. Click Add Conditions.
6. Make sure that the Predicate List drop-down shows Group.
7. Click Next.
8. Enter the group name in the GroupArgument Name field, as in Table 3–3.
9. Click Add.
10. Click Finish.
11. Click Save.
12. Go back to the Global Roles list by clicking Global Roles in the list of bread-crumbs at the top of the screen.

3.1.5.5 Creating the Oracle RTD Administrator and Other Users

To create an Oracle RTD Administrator, perform the following steps:

1. Go to the myrealm page, and select the Users and Groups tab.
2. Select the Users sub-tab.
3. Create an Oracle RTD administrator.
4. Click the New button.
5. Enter the user's name and password.
6. Click OK.
7. Click the user name, to edit the user properties.
8. Select the Groups tab.
9. Select the RTDAdminGroup in the left list, Available, and click the right arrow to move it to the right list, Chosen.
10. Click Save.
11. Return to the Users and Groups tab by selecting Users and Groups in the list of bread-crumbs at the top of the screen.

To create more users and put them into appropriate groups referenced by the roles, perform steps 4 through 11 of the preceding list of steps for each user.

3.1.6 Deploying Oracle RTD to WebLogic

For general information about deploying WebLogic applications, see *Oracle Fusion Middleware Deploying Applications to Oracle WebLogic Server*.

To deploy Oracle RTD to your cluster, perform the following steps:

1. Logon to the Administration Console:

   `http://weblogic_host:port/console`

2. In the tree to the left under Domain Structure, navigate to Deployments > Install.
3. In the Install Application Assistant page, set the Path of the deployment to the location of deployment to the `expanded` folder that you set up in Section 2.1, "Installing Oracle Real-Time Decisions Files."

For example, `RTD_HOME\package\expanded`.

4. Set **Install this deployment as an application**.
5. Select Cluster for the deployment target.
6. Set the Name as OracleRTD.
7. In the Security area, select **Custom Roles**.
8. In the Source accessibility area, select **Copy this application onto every target for me** under **Recommended selection**.
9. Select the cluster as the target on which to deploy.

   After deployment, the OracleRTD application is in a Prepared state.
10. Start the OracleRTD application..

    This converts the application state from Prepared to Active.

    The Oracle RTD log file can be found at `RTD_RUNTIME_HOME/log/server.log`.

    For example

    - `MW_HOME/user_projects/domains/domain_name/servers/server_name/stage/OracleRTD/expanded/log/server.log`

3.1.7 Restarting WebLogic Server

This process consists of restarting the Admin Server and Node Manager, then starting the server or servers in the cluster.

To restart the Admin Server and Node Managers, perform the following steps:

1. Stop Node Manager.

   If Node Manager has been set up as a web service, you can stop Node Manager from the Services window.

   Alternatively, find the command window where the Node Manager has been started (from a previous execution of `startNodeManager.cmd`) and hit Ctrl-C or close the window.
2. Stop the Admin Server.

   Find the command window where the Admin Server has been started (from a previous execution of `startWebLogic.cmd`) and hit Ctrl-C or close the window.

   Alternatively, run the command:

   - `weblogic-install-dir/user_projects/domains/rtd_domain/bin/stopWebLogic.cmd`

   For example:

   - `c:/Oracle/Middleware/user_projects/domains/rtd_domain/bin/stopWebLogic.cmd`
3. Start the AdminServer.

   Run the command:

   - `weblogic-install-dir/user_projects/domains/rtd_domain/bin/startWebLogic.cmd`

   For example:

   - `c:/Oracle/Middleware/user_projects/domains/rtd_domain/bin/startWebLogic.cmd`

4. Start Node Manager.

   If Node Manager has been set up as a web service, you can start Node Manager from the Services window.

   Alternatively, run the command:

   - `weblogic-install-dir\wlserver_10.3\server\bin\startNodeManager.cmd`

For a typical installation, the log files for AdminServer can be found at:

- `C:\Oracle\Middleware\user_projects\domains\rtd_domain\servers\AdminServer\logs\`

3.1.8 Starting Clustered Servers

If the cluster servers have not been automatically started as part of the WebLogic Server restart, perform the following steps in the WebLogic Administration Console to start the clustered servers:

1. In the tree to the left under Domain Structure, expand **Environment** and click **Clusters**.
2. Click the cluster name whose servers you want to start.
3. Click the Control tab.
4. Check all servers in the cluster.
5. Click **Start**.
6. Click **Yes**.
7. Wait until the installation finishes.

For a typical installation, the log files for the managed server MS1 can be found at:

- `C:\Oracle\Middleware\user_projects\domains\rtd_domain\servers\MS1\logs\`

3.1.9 Starting OracleRTD

If the OracleRTD application has not been automatically started as part of the WebLogic Server restart, perform the following steps in the WebLogic Administration Console to start the OracleRTD application:

1. In the tree to the left under Domain Structure, click **Deployments**.
2. Check the **OracleRTD** checkbox.
3. Click **Start** and select **Servicing all Requests**.
4. Click **Yes**.
5. Wait until the installation finishes.

For a typical installation, the OracleRTD log file `server.log` can be found at:

- `C:\Oracle\Middleware\user_projects\domains\rtd_domain\servers\MS1\stage\OracleRTD\expanded\log\`

3.2 Verification Tasks

This section describes a number of system verification processes, which you can perform after installation and after any significant system enhancement or modification, such as described in Section 3.3, "Post-Installation Tasks."

Verify Oracle RTD Deployment

- Section 3.2.1, "Viewing OracleRTD MBeans in JConsole"
- Section 3.2.2, "Deploying an Inline Service in Decision Studio"
- Section 3.2.3, "Running Integration Point Tests in CrossSell Inline Service"
- Section 3.2.4, "Running Load Generator Against CrossSell Inline Service"
- Section 3.2.5, "Viewing Reports in Decision Center"

3.2.1 Viewing OracleRTD MBeans in JConsole

To view the MBeans for the OracleRTD application in JConsole, perform the following steps:

1. Run `jconsole.exe` from a Java installation.

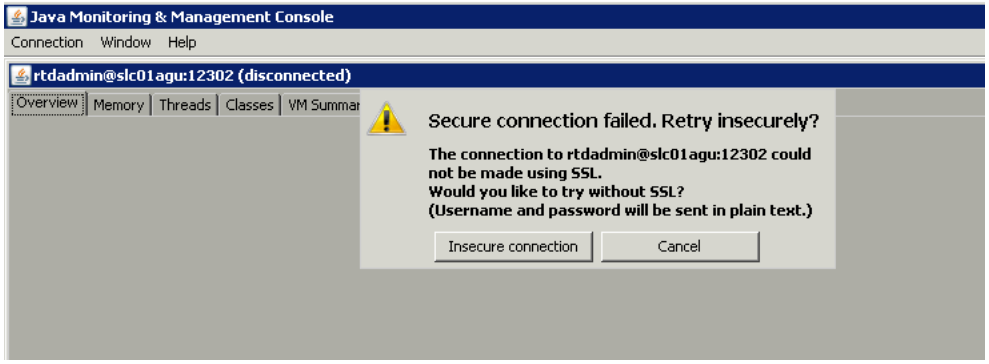
   For example: `C:\Program Files\Java\jdk1.7.x\bin\jconsole.exe`

2. In the New Connection window, click Remote Process.
3. In the unnamed box under Remote Process, enter host and port values in the format <*hostname*>:<*port*>.

   The host is the machine name of the first clustered server instance, and the port is the JMX remote port set for this server instance as specified in Section 3.1.3, "Configuring JVM Properties for Managed Server."

   For example: `srvmach01:12302`

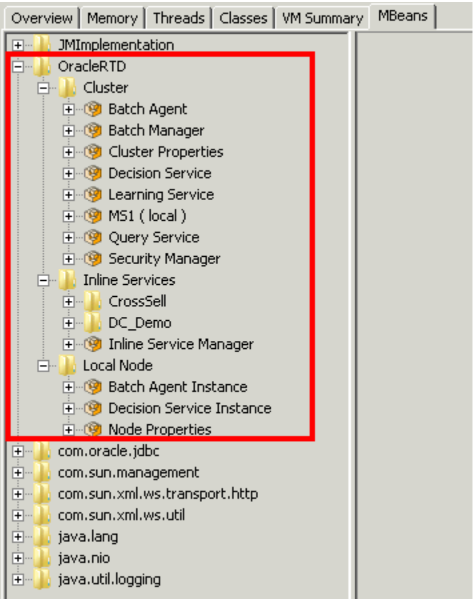
4. Enter the administrator username and password.
5. Click **Connect**.
6. If you are not using SSL, click Insecure connection.



Secure connection failed. Retry insecurely?

The connection to rtdadmin@slc01agu:12302 could not be made using SSL.
Would you like to try without SSL?
(Username and password will be sent in plain text.)

You can view the OracleRTD MBeans entries, similar to the following:



Note: *The preceding image shows Inline Services entries. Inline Service entries only appear only after the Inline Services have been deployed.*

3.2.2 Deploying an Inline Service in Decision Studio

If you are running Decision Studio on a 32 bit machine, go to the client computer where you installed the Oracle RTD client tools and run

`RTD_HOME\eclipse\eclipse.exe`.

To run Decision Studio on a 64 bit machine, you must specify a 32 bit JDK on the command line, in the following format:

```
RTD_HOME\eclipse\eclipse.exe -vm "32bit_jdk_directory\bin\javaw"
```

For example:

```
C:\OracleBI\RTD\eclipse\eclipse.exe -vm "C:\Program Files
(x86)\Java\jdk1.7.x\bin\javaw"
```

Note: The following assumes that you have set up the tables for the CrossSell Inline Service, as described in Section 2.3, "Populating the CrossSell Example Data (Optional)." If you have not done this, set up and deploy your own Inline Service.

Import the CrossSell Inline Service as an existing project from `RTD_HOME\RTD\examples\CrossSell\`.

Deploy the CrossSell Inline Service, with Host= your host machine, and Port=either your managed server port (or 8080 if you have set up the web server).

3.2.3 Running Integration Point Tests in CrossSell Inline Service

Before you can perform tests on the integration points in the CrossSell Inline Service, you must have first deployed the CrossSell Inline Service, as described in Section 3.2.2, "Deploying an Inline Service in Decision Studio."

To test the integration points in the CrossSell Inline Service, perform the following steps:

1. In the CrossSell Inline Service in Decision Studio, click the Test tab.
2. **Call Start**
   a. In the Test tab, select the Call Start Integration Point.
   b. Enter 5 for Customer ID and "Cell" for channel.
   c. Click the (right-arrow) Execute Request icon.
3. Offer Request
   a. In the Test tab, select the Offer Request Integration Point.
   b. Enter 5 for Customer ID and "Cell" for channel.
   c. Click the (right-arrow) Execute Request icon.
   d. Note which offer is generated in the Response tab.
4. **Offer Response**
   a. In the Test tab, select the Offer Response Integration Point.
   b. Enter 5 for Customer ID and "Cell" for channel.
   c. For choiceName, enter the offer noted in 3d.
   d. Enter "Interested" for choiceOutcome and "off" for simulationMode.
   e. Click the (right-arrow) Execute Request icon.
5. **Call Resolution**
   a. In the Test tab, select the Call Resolution Integration Point.
   b. Enter 5 for Customer ID and "Cell" for channel.
   c. Click the (right-arrow) Execute Request icon.

Note: If an error message appears in the Log tab indicating that a Decision Service request was received from an untrusted host, you must enter the IP address of your host in JConsole, in

- `MBeans > OracleRTD > Cluster > Decision Service > TrustedDSClients`

This change takes effect immediately, that is, you do not have to restart the cluster or managed server.

3.2.4 Running Load Generator Against CrossSell Inline Service

Note: The following assumes that you have set up the tables for the CrossSell Inline Service, as described in Section 2.3, "Populating the CrossSell Example Data (Optional)." If you have not done this, run the Load Generator against your own Inline Service.

To run Load Generator against the CrossSell Inline Service, perform the following steps:

1. If you have not set up the web server, edit the file *`RTD_HOME`*`\client\clientHttpEndPoints.properties` as follows:

   Change the port number on the line

   - `HTTP1.url = http://localhost:`**`8080`**

   to the port of the managed server, for example,

   - `HTTP1.url = http://localhost:`**`7003`**

2. Start Oracle RTD Load Generator by running:

   `RTD_HOME``\scripts\loadgen.cmd`

3. Click **Open an existing Load Generator script**.
4. Go to *`RTD_HOME`*`\examples\CrossSell\etc` and select `LoadGen3Threads.xml`.
5. Click **Open**.
6. Click the General tab.
7. Next to **Client Configuration File**, on the right, click the ellipsis (...) button.
8. Go to *`RTD_HOME`*`\client`.

   Select **clientHttpEndPoints.properties**.

9. Click **Open**.
10. In the top menu bar, click **File,** then **Save**.
11. In the top menu bar, click the icon **Runs the current Load Generator script** (blue right arrow).

3.2.5 Viewing Reports in Decision Center

To view reports in Decision Center, perform the following steps:

1. In a browser window, enter `http://`*`hostname`*`:`*`managed_server_port`*`/ui`

   For example, `http://localhost:7003/ui`.

2. Enter the Oracle RTD administrator user name and password, then click Login.
3. Open the CrossSell Inline Service.
4. In the left-hand menu, click Offers.
5. Select different choices, and explore the reports for the choices.

   For example, Performance > Counts, Analysis > Best-fit, Analysis > Quality > Cumulative Gains Chart.

Note: Depending on which choices were generated by Load Generator, some choices may have fuller reports than others.

3.3 Post-Installation Tasks

This section describes the tasks that you can perform after a basic system setup to add more features and capabilities into your system.

After each post-installation task, you can optionally perform one or more of the basic system verification tasks described in Section 3.2, "Verification Tasks."

This section contains the following topics:

- Section 3.3.1, "Setting Up and Configuring the Web Server"
- Section 3.3.2, "Setting Up JConsole Security for WebLogic (Recommended)"
- Section 3.3.3, "Configuring Additional JDBC Sources"
- Section 3.3.4, "Scaling Up"
- Section 3.3.5, "Scaling Out"

3.3.1 Setting Up and Configuring the Web Server

Note: This section describes a basic Web Server setup for a WebLogic system. For more details and further information, see *Configuring Web Server Functionality* in *Oracle Fusion Middleware Configuring Server Environments for Oracle WebLogic Server*.

This process consists of first installing Apache *(download it from the Web)*, then enabling the WebLogic proxy plug-in in the servers, then configuring up the Apache HTTP server.

3.3.1.1 Enabling WebLogic Proxy Plug-In in Servers

To enable the WebLogic proxy plug-in in the servers, perform the following steps in the WebLogic Administration Console:

1. In the tree to the left under Change Center, click **Lock & Edit**.
2. In the tree to the left under Domain Structure, expand **Environment** and click **Servers**.
3. Click the managed server name.
4. Under Configuration tab General section, click **Advanced**.
5. Under Advanced, check **WebLogic Plug-In Enabled** checkbox.
6. Click **Save**.
7. In the tree to the left under Change Center, click **Activate Changes**.

3.3.1.2 Configuring Apache HTTP Server

After installing Apache, to configure the Apache HTTP server, perform the following steps:

1. Copy
   - `weblogic-install-dir\wlserver_10.3\server\plugin\win\32\mod_wl_22.so`

   to

   - `apache-install-dir\modules\`

   For example:

   ```
   C:\Program Files\Apache Software Foundation
   \Apache2.2\modules\
   ```
2. Open `apache-install-dir\conf\httpd.conf`.

For example:

`C:\Program Files\Apache Software Foundation \Apache2.2\conf\httpd.conf`

(Alternatively: **Start>All Programs>Apache HTTP Server 2.2 >Configure Apache Server>Edit the Apache httpd.conf Configuration File**)

3. Change the port for property **Listen** to 8080.
4. Change the port for property **ServerName** to 8080.

   For example: `ServerName myHostName.mynetwork.com:8080`

5. At the end of `httpd.conf`, add the following lines, after replacing hostName* and 7002-7007 in the `WebLogicCluster` line with the server and port for your system.

   [Note that you can include servers and ports that you intend to create in the future, as in the following example, where extra servers on srvmach01 with ports 7004 and 7005 may not have been set up yet.]

```
LoadModule weblogic_module modules/mod_wl_22.so
<IfModule mod_weblogic.c>
    # RTDCluster_DsDcLs
    <LocationMatch "/(ui|schema|soap|rtis|ls)">
      SetHandler weblogic-handler
      WebLogicCluster srvmach01:7003,srvmach01:7004,srvmach01:7005
    </LocationMatch>
 </IfModule>
```

6. Save `httpd.conf`.
7. Start Apache HTTP: **Start>All Programs>Apache HTTP Server 2.2 >Control Apache Server>Start Apache in Console**

   (Or `C:\Program Files\Apache Software Foundation\Apache2.2\bin>httpd -k start`)

   Note: To stop, close the window in which you started Apache.

   (Or `C:\Program Files\Apache Software Foundation\Apache2.2\bin>httpd -k stop`)

3.3.2 Setting Up JConsole Security for WebLogic (Recommended)

Perform the following steps to set up security for the JConsole management tool.

1. Navigate the path:

   Environment > Servers > *managed_server_name* > Configuration > Start Server tab.

2. If you have not set the `-Dcom.sun.management.jmxremote.authenticate` parameter to "true" during the previous process of Configuring JVM Properties for Managed Server, then, for the Arguments field, add the following, all on one line:

```
-Dcom.sun.management.jmxremote.authenticate=true
```

3. Save.
4. Go to *JAVA_HOME*`/jdkversion/jre/lib/management`. Then, create a copy of the `jmxremote.password.template` file and rename it to `jmxremote.password`.

5. Open the `jmxremote.password` file for editing. At the end of the file, remove the pound character (#) in front of the following two lines:

   ```
   monitorRole QED
   controlRole R&D
   ```

   Each line lists a set of credentials, or in other words, a user name followed by the corresponding password. Optionally, you can add new user names and passwords on separate lines. If you decide to keep the default user names `monitorRole` and `controlRole`, change the default passwords to new ones of your choice.

   By default, the user name `monitorRole` allows JConsole MBean read-only permissions, while the user name `controlRole` allows for full JMX MBean control. The following step explains how to change these default permissions.

6. To set permissions for each set of credentials, open the `jmxremote.access` file, located in the same directory, for editing. Then, use the keywords readonly and readwrite to specify the access level for each set of credentials. For example:

   ```
   monitorRole readonly
   controlRole readwrite
   ```

   Make sure to add permissions for any new credentials you added to the `jmxremote.password` file.

7. Because the `jmxremote.password` file contains passwords in clear text, you need to restrict access to this file to the file owner by changing the file access permissions, as follows:

   - **On Windows:** Follow the instructions at:

     `http://docs.oracle.com/javase/7/docs/technotes/guides/management/security-windows.html`

   - **On UNIX:** Run the command `chmod 600 jmxremote.password`

   For more information on JConsole security, go to:

   `http://docs.oracle.com/javase/7/docs/technotes/guides/management/agent.html`

   Important: You *must* change the file access permissions for the `jmxremote.password` file. Do not skip this step.

8. Restart the managed server.

You can now run JConsole and log in, using the user name and password combination defined in the `jmxremote.password` file. See Section 11.3, "Accessing JConsole" for more information.

3.3.3 Configuring Additional JDBC Sources

To configure an additional JDBC data source, perform the following steps:

1. In the tree to the left under Change Center, click **Lock & Edit.**
2. Navigate the path **Services** -> **Data Sources** -> **New** -> **Generic Data Source**.
3. For **Name**, enter the name for the data source.
4. For **JNDI Name**, enter any name (except SDDS).

5. For **Database Type**, select **Oracle**.
6. Click **Next**.
7. For **Database Driver**, select **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later**.
8. Click **Next**.
9. Uncheck the **Supports Global Transactions** checkbox.
10. Click **Next**.
11. On the Connection Properties page, perform these steps:
    a. For **Database Name**, enter the name of the database referenced by the data source.
    b. For **Host Name**, enter the name of the computer hosting the database server.
    c. For **Port**, enter the port number on the database server used to connect to the database.
    d. For **Database User Name**, enter the name of the database run-time user.
    e. For **Password**, enter the password of the database run-time user, then click **Next**.
12. On the Test Database Connection page, leave all the settings already filled, except, for **Test Table Name**, enter a table in the database.
13. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**..
14. Under Clusters, select the checkbox for your cluster.
15. Click **Finish**.

    (Optional) If you want to improve performance, you should consider turning off the "Wrap Data Types" option, by performing these steps:
    a. Select the data source that you just created.
    b. Under the Configuration tab, click on the Connection Pool sub tab.
    c. Scroll down and expand the Advanced section.
    d. Deselect **Wrap Data Types**.
    e. Click **Save**.
16. In the tree to the left under Change Center, click **Activate Changes**.

3.3.4 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, then modifying their Server Start parameters.

> **Note:** As the OracleRTD application is targeted to a cluster, you do not have to deploy OracleRTD separately on each additional server.

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

Then add JVM and startup arguments to the new managed server. To do this, perform essentially the same steps described for creating the first managed server in Section 3.1.3, "Configuring JVM Properties for Managed Server."

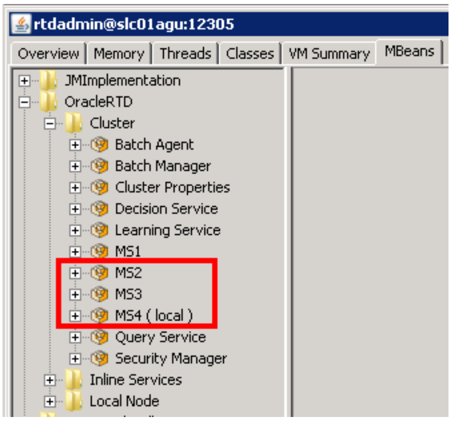
As for the first managed server, for each extra server, you must provide server-specific values for the following Server Start parameters:

- **rtd.instanceName** (specify the managed server name)
- **rtd.HTTPHostAndPort** (specify host and port for the managed server)
- **com.sun.management.jmxremote.port** (specify the port for JConsole)

After you save your changes, activate them.

Then start the new managed server.

When all the new managed server startups have completed successfully, you can log into JConsole (using any of the managed server jmxremote.port values), and see the new managed servers in the expanded list of MBeans under `OracleRTD > Cluster`, such as in the following example:



3.3.5 Scaling Out

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

1. Install WebLogic on the second (target) machine.
2. Start Node Manager on the target machine.
3. On the target machine:
    - Start the WebLogic Scripting Tool (`wlst.sh` or `wlst.cmd`)
    - Connect to the first (source) machine
    - Enroll the second (target) machine in the existing domain *(use `nmEnroll`)*
    - Disconnect from the first machine
    - Quit the WebLogic Scripting Tool
4. On the source machine:
    - Clone the existing Machine

      For the cloned Machine, specify a new name, and new host and IP address values

- Clone the existing server and specify that the new server is on the cloned Machine

5. Start the cloned server.

3.4 Uninstalling the Oracle Real-Time Decisions Application from WebLogic

You can use the WebLogic Server Administration Console to uninstall Oracle RTD from WebLogic. Before you begin, ensure that WebLogic is started.

To uninstall Oracle RTD from WebLogic:

1. Access the WebLogic Server Administration Console for your Oracle RTD domain at the URL `http://`*`weblogic_host`*`:`*`port`*`/console`. At the login prompt, enter the administrator user name and password. On Windows, you can also access the WebLogic Server Administration Console through **Start > Programs > Oracle WebLogic > User Projects >** ***domain_name*** **> Admin Server Console**.
2. In the tree on the left, click **Deployments**.
3. Select **OracleRTD** and click **Stop > Force Stop Now**.
4. Select **OracleRTD** and click **Delete**.

These steps uninstall Oracle RTD from WebLogic, but they do not remove the Oracle RTD files from the operating system.

To be able to redeploy Oracle RTD, you must delete the Oracle RTD files manually, with the managed server shut down. Delete the `apps` folder under the directory into which you expanded the file `RTD.ear` when you installed Oracle RTD. For more information about this directory, see Step 2 of Section 2.1.1, "Installing Oracle RTD on a Single Windows Computer" or Step 2 of Section 2.1.2, "Installing Real-Time Decision Server on UNIX."

3.5 Configuring SSL for Real-Time Decision Server (Recommended)

Follow the steps in this section to set up SSL for all client connections to Real-Time Decision Server. Before you begin, ensure that you followed the instructions in Section 2.5 to change the default Oracle RTD keystore and truststore passwords. Also, ensure that WebLogic is started.

Note: If you want to use your own keystore and truststore, you do not need to complete the instructions in Section 2.5.

To configure SSL for Real-Time Decision Server:

1. Access the WebLogic Server Administration Console for your Oracle RTD domain at the URL `https://`*`weblogic_host`*`:`*`port`*`/console`. At the login prompt, enter the administrator user name and password. On Windows, you can also access the WebLogic Server Administration Console through **Start > Programs > Oracle WebLogic > User Projects >** ***domain_name*** **> Admin Server Console**.
2. In the tree on the left, expand **Environment** and choose **Servers**.
3. Click the name of the RTD Managed Server (for example, **RTD_Server**).
4. In the **General** tab, select **SSL Listen Port Enabled**, then enter `8443` for **SSL Listen Port**.

5. Click the **Keystores** tab, then provide the following values:
   a. For **Keystores**, select **Custom Identity and Custom Trust**.
   b. For **Custom Identity Keystore**, enter *RTD_HOME*`/etc/ssl/sdserver.keystore`. Alternatively, if you do not want to use the default Oracle RTD keystore, enter the path to your own keystore.
   c. For **Custom Identity Keystore Type**, enter `JKS`.
   d. For **Custom Identity Keystore Passphrase** and **Confirm Custom Identity Keystore Passphrase**, enter the password for your keystore. If you are using the default Oracle RTD keystore, enter the password you created in Section 2.5.
   e. For **Custom Trust Keystore**, enter *RTD_HOME*`/etc/ssl/sdtrust.store`. Alternatively, if you do not want to use the default Oracle RTD truststore, enter the path to your own truststore.
   f. For **Custom Trust Keystore Type**, enter `JKS`.
   g. For **Custom Trust Keystore Passphrase** and **Confirm Custom Trust Keystore Passphrase**, enter the password for your truststore. If you are using the default Oracle RTD truststore, enter the password you created in Section 2.5.
6. Click the SSL tab, then provide the following values:
   a. For **Identity and Trust Locations**, select `Keystores`.
   b. For **Private Key Alias**, enter `tc-ssl`.

   For **Private Key Passphrase** and **Confirm Private Key Passphrase**, enter the password for your keystore. If you are using the default Oracle RTD keystore, enter the password you created in Section 2.5.
7. Click **Save**.
8. Restart the RTD Managed Server.

> **Note:** For a truly secure environment, you should also disable the regular HTTP port to ensure that all client connections are routed through the SSL port. To do this, perform the following step:
>
> 1. Disable the HTTP port for your Web server using application server tools. Refer to the WebLogic documentation for more information.

9. If you are using your own keystore and truststore, perform the following additional steps to enable SSL for Decision Studio. You do not need to perform these steps if you are using the default Oracle RTD keystore and truststore.
   a. Open *RTD_HOME*`\eclipse\eclipse.ini` for editing.
   b. Locate the following line:

   ```
   -Djava.net.ssl.trustStore="..\etc\ssl\sdtrust.store"
   ```

   c. Replace `..\etc\ssl\sdtruststore` with the full path to your truststore file.
   d. Save and close the file.
   e. Open *RTD_HOME*`\scripts\sdexec.cmd` for editing.

f. Locate the line beginning with `%SD_START%`, near the bottom of the file. Near the end of the line, locate the following string:

```
-Djavax.net.ssl.trustStore="%SD_ROOT%\etc\ssl\sdtrust.store"
```

g. Replace `%SD_ROOT%\etc\ssl\sdtruststore` with the full path to your truststore file.

h. Save and close the file.

3.5.1 Testing the SSL Configuration

To verify that the SSL port is functioning properly, go to Decision Center at the URL `https://`*`server_name`*`:`*`ssl_port`*`/ui`. If the SSL port is functioning property, your browser will display the "Welcome to Decision Center" login screen.

You may get a message from your Web browser, similar to "Do you want to accept this certificate?" This message is generated because the browser does not know about the self-signed certificate that was shipped with the default Oracle RTD keystore. This self-signed certificate is suitable for development and test environments, but it is not recommended for production environments.

For production environments, Oracle recommends the self-signed certificate be replaced with a certificate from a trusted certificate authority (CA), like Verisign/Thawte, by submitting to the CA a certificate request generated by Sun's keytool utility.

For instructions on generating a certificate request, and for importing the certificate from the CA into the keystore, go to one of the following URLs:

For Linux and Solaris:

http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html

For Windows:

http://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html

4

Installing and Configuring WebSphere for Oracle Real-Time Decisions

This chapter describes how to install and configure Oracle RTD on WebSphere.

The process of installing and configuring WebSphere for Oracle RTD consists of four major stages:

- WebSphere installation.

  This consists of installing everything that is required for a basic WebSphere cluster, including the HTTP server (software load balancer) and all WebSphere Fixpacks.

- WebSphere profile creation.

  This consists of creating a profile containing a deployment manager and a managed node.

  A subsequent step, after the initial installation, describes how to expand this setup to include an additional managed node.

- WebSphere configuration.

  After the profile is created, you must configure certain parts of that profile to make it ready for Oracle RTD deployment. This includes setting server properties, JDBC elements and configuring HTTP plugins.

- Oracle RTD deployment and post-deployment steps.

This chapter contains the following sections:

- Section 4.11, "Configuring New Data Sources"
- Section 4.12, "Configuring SSL for Real-Time Decision Server (Recommended)"

4.1 Installing WebSphere Application Server and Associated Components

This section describes how to install WebSphere Application Server and associated components:

- IBM Installation Manager
- WebSphere Application Server Network Deployment
- IBM HTTP Server
- Web Server Plug-ins for WebSphere Application Server
- WebSphere Customization Toolbox

Preparing /Prerequisites

The operating-system dependent installation source files consist of several sets of zip files, for both the base source files and associated Fixpacks:

- IBM Installation Manager
- WebSphere 8 Application Server Network Deployment
- WebSphere 8 Supplements (including HTTP Server and Plug-in)
- WebSphere 8 Fixpacks for the different components:
    - Fixpacks for WebSphere Application Server Network Deployment
    - Fixpacks for WebSphere Application Server Supplements
    - Fixpacks for WebSphere Customization Toolbox

Note: There may also be other Fixpacks for interim fixes as recommended by IBM for your WebSphere version.

Unzip the installation source files for the "base level" software components (and combine them when there are several zip files per software component) into temporary directories, using the following table as a guideline:

Table 4–1 IBM Base Level Software Components

Software Component	Suggested Temporary Directory
IBM Installation Manager	WAS8-InstallManager
WebSphere Application Server 8.0 Network Deployment	WAS8-Main
WebSphere Application Server 8.0 Supplements	WAS8-Supplements

Unzip the installation source files for the Fixpacks into temporary directories, using the following table as a guideline:

Note: *It is convenient to incorporate the latest fix pack level in the directory names. At the time of writing, the latest Fixpack level was 8.0.0.5.*

Table 4–2 Fix Packs for Websphere Application Server 8

Latest Fix Packs for...	Suggested Temporary Directory
WebSphere Application Server Network Deployment	WASFP8005AppServerND
WebSphere Application Server Supplements	WASFP8005Supplements
WebSphere Customization Toolbox	WASFP8005CustomToolbox

To install the IBM Installation Manager

Run the `install` command (if you have the necessary administrative rights on your target machine) or `userinst` from the IBM Installation Manager temporary directory.

This installs IBM Installation Manager into the Installation Manager directory *(with default lowest levels /IBM/InstallationManager/eclipse)*, from which you can install other IBM software components.

At the end of the installation process, click **Restart Installation Manager** to start the process of installing IBM WebSphere Application Server and supplementary software components.

To install the IBM WebSphere Application Server and Components

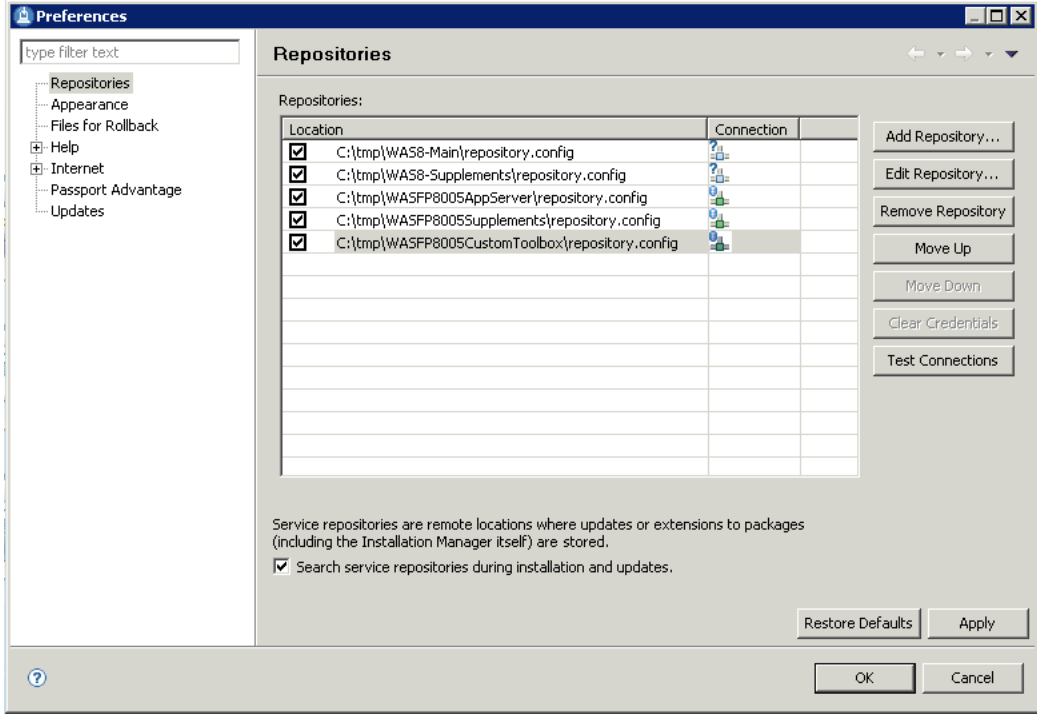
This process installs the following products:

- WebSphere Application Server Network Deployment
- IBM HTTP Server
- Web Server Plug-ins for WebSphere Application Server
- WebSphere Customization Toolbox

Installing WebSphere Application Server and Associated Components

1. If you have not already started the Installation Manager, launch it from the Installation Manager directory.
2. Click **Install**.
3. Click the **Repositories** link to start configuring the repositories for installing the WebSphere Application Server and the associated components.
4. For each of the software components in Table 4–1 except IBM Installation Manager and for each of the fix packs in Table 4–2, perform the following steps:
   a. Click **Add Repository...**
   b. Browse to the temporary directory for the software component or fixed pack.
   c. Select the `repository.config` file.

   The resultant display is similar to the following:



Repositories

Repositories:

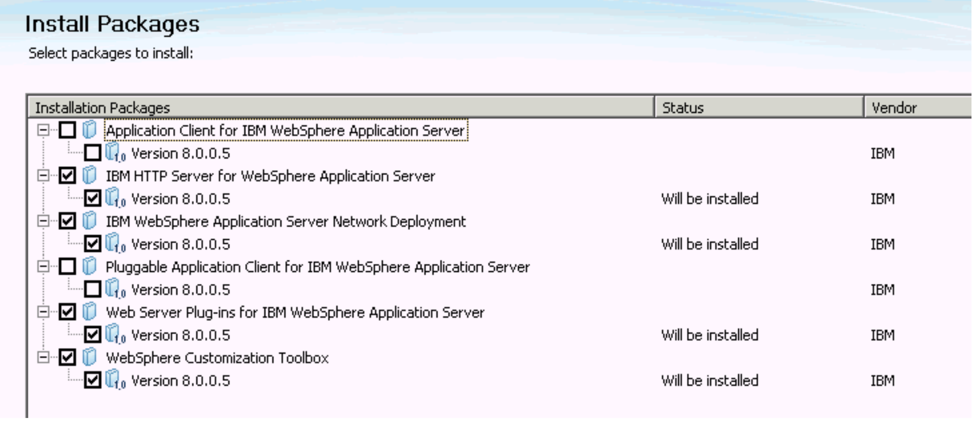
Location	Connection
☑ C:\tmp\WAS8-Main\repository.config	
☑ C:\tmp\WAS8-Supplements\repository.config	
☑ C:\tmp\WASFP8005AppServer\repository.config	
☑ C:\tmp\WASFP8005Supplements\repository.config	
☑ C:\tmp\WASFP8005CustomToolbox\repository.config	

Service repositories are remote locations where updates or extensions to packages (including the Installation Manager itself) are stored.

☑ Search service repositories during installation and updates.

5. Click **OK**.

6. In the Install Packages screen, select the packages for the four components that you want to install.



Install Packages

Select packages to install:

Installation Packages	Status	Vendor
☐ Application Client for IBM WebSphere Application Server		
☐ Version 8.0.0.5		IBM
☑ IBM HTTP Server for WebSphere Application Server		
☑ Version 8.0.0.5	Will be installed	IBM
☑ IBM WebSphere Application Server Network Deployment		
☑ Version 8.0.0.5	Will be installed	IBM
☐ Pluggable Application Client for IBM WebSphere Application Server		
☐ Version 8.0.0.5		IBM
☑ Web Server Plug-ins for IBM WebSphere Application Server		
☑ Version 8.0.0.5	Will be installed	IBM
☑ WebSphere Customization Toolbox		
☑ Version 8.0.0.5	Will be installed	IBM

Note: The components required for the cluster and Oracle RTD are:

- WebSphere Application Server Network Deployment
- IBM HTTP Server (to act as the cluster load balancer)
- Web Server Plug-in (to make HTTP Server and Application Server Network Deployment work together)
- WebSphere Customization Toolbox (to create profiles and to configure the Web Server Plug-ins)

---

7. Click **Next**.

8. If you have previously unzipped and selected repositories for extra fixes, select those fixes, and click **Next**.
9. Read and then accept the License agreement. Click **Next**.
10. In the Shared Resources Directory field, click **Browse** and select the directory for shared resources. Click **Next**.
11. You can select each package and change its Installation Directory field. Oracle recommends that you install all components into the same base directory. Then click **Next**.

> **Note:** In this section and subsequent sections, it is assumed that all your components are installed under one base directory, which is referred to as *WAS_HOME*.

12. Select the translation that you wish to install, then click **Next**.
13. In the Features screen, expand the packages, then select and deselect the features for your particular system.

    For example, deselect the Customization Toolbox options for z/OS unless you require them.

    Oracle recommends that all the SDK and Java options are consistent, that is, all 32-bit or all 64-bit.

    Click **Next**.
14. When you are prompted for the IBM HTTP Server configuration parameters:
    a. Change the HTTP Port to 8080.
    b. On Windows machines, select whether to run IBM HTTP Server as a Windows service. If you select this option, you must also specify whether the service should start automatically or manually after a system reboot.
    c. Click **Next**.
15. Review your selections, then click **Install**.

    IBM Installation Manager installs the packages that you selected in step 6.
16. When the installation process completes successfully, you are prompted to start the profile management tool to create profiles.

    Select **Profile Management Tool to create a profile** and click **Finish**.

    The WebSphere Customization Toolbox appears.

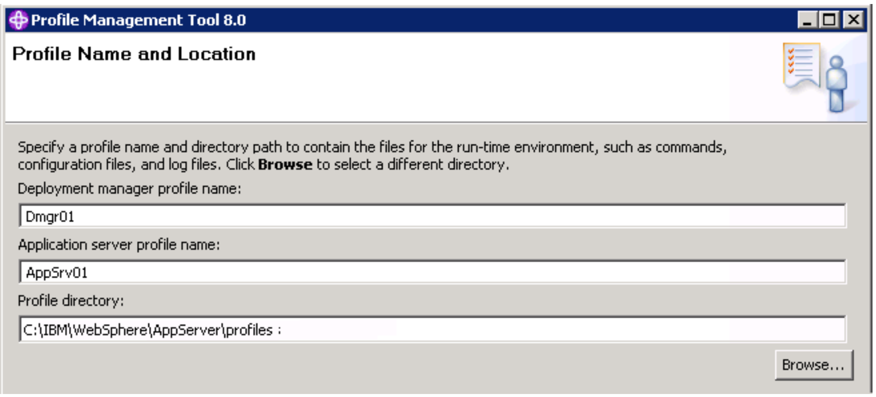
Creating a Profile

> **Note:** If you do not create the profile during your installation session, you can launch WebSphere Customization Toolbox using the script `pmt.sh` or `pmt.bat` under *WAS_HOME*/`WebSphere/AppServer/bin/ProfileManagement`.

1. In the WebSphere Customization Toolbox, click **Create...**
2. In the Environment Selection screen, select **Cell (deployment manager and a federated application server)**. The Cell environment creates a deployment

manager node, a regular managed node and one managed server on that node. The server that is created is called "server1", but will not be required, and will be manually deleted in a subsequent step. Click **Next**.

3. In the Profile Creation Options screen, select **Advanced profile creation** *(this enables you to use default configuration settings or to specify your own values)*, then click **Next**.
4. In the Optional Application Deployment screen, select **Deploy the administrative console (recommended)**, then click Next.
5. In the Profile Name and Location screen, you can change the names of the profiles and profile directory.



Profile Name and Location

Specify a profile name and directory path to contain the files for the run-time environment, such as commands, configuration files, and log files. Click **Browse** to select a different directory.

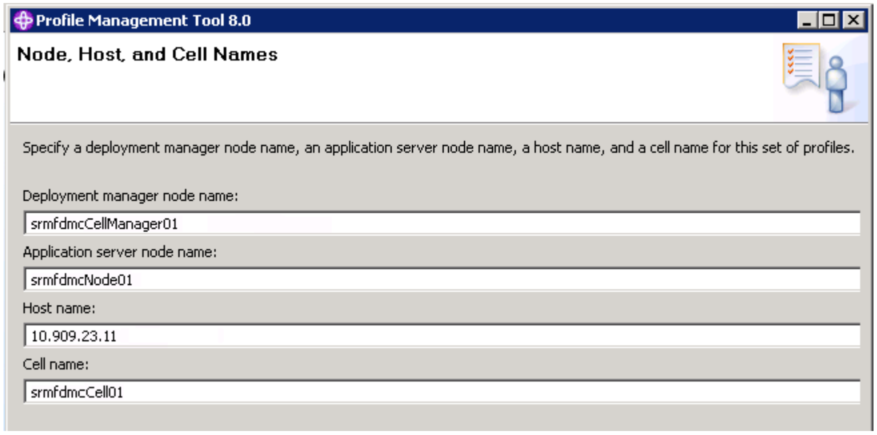
Deployment manager profile name: Dmgr01

Application server profile name: AppSrv01

Profile directory: C:\IBM\WebSphere\AppServer\profiles :

Click **Next**.

6. In the Node, Host, and Cell Names screen, you can change the names of the nodes, host, and cell. You can leave the Host Name as the default DNS name, or you can set it to an IP address. If you select a DNS name, make sure that it is correctly listed and matched with an IP address in /etc/hosts.



Node, Host, and Cell Names

Specify a deployment manager node name, an application server node name, a host name, and a cell name for this set of profiles.

Deployment manager node name: srmfdmcCellManager01

Application server node name: srmfdmcNode01

Host name: 10.909.23.11

Cell name: srmfdmcCell01

Click **Next**.

7. In the Administrative Security screen, select **Enable administrative security**, enter the security username and password for the Deployment Manager console, then click **Next**.
8. In the Security Certificate screens, enter certificate and keystore information as required, then click **Next**.
9. In the Port Values Assignment screens, you can change the default values as required for your system, then click **Next**.

Note: The most significant ports for subsequent administrative tasks are:

- WC_defaulthost - this is where the server listens for incoming HTTP traffic (when you create your own servers, this is the port used to check if they respond).
- WC_adminhost_secure - this is where the administrative console is to be found.
- SOAP_CONNECTOR_ADDRESS - this is where the deployment manager waits for commands (for example, it is used when you subsequently extend of the cluster).

---

10. On Windows machines, in the Windows Services Definition screen, you can select the following:
    - Whether to run the deployment manager as a Windows service
    - The logon option (local system account or specified user account)
    - The startup type (manual or automatic)

    Click **Next**.

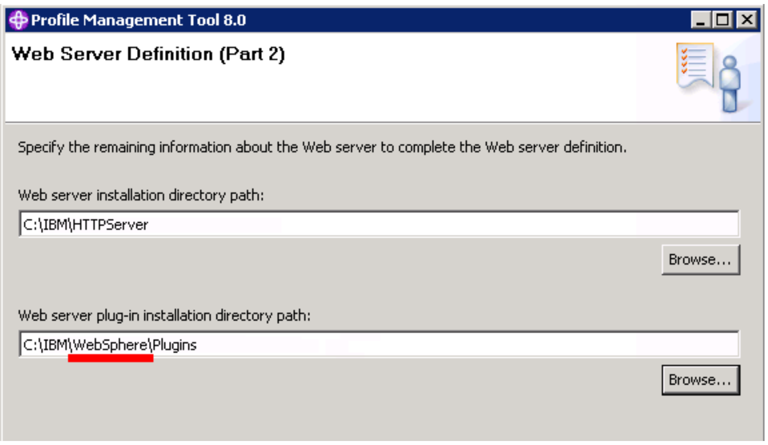
11. As IBM HTTP Server has already been installed, you can now define a web server in the Web Server Definition screen.

    Specify names and port as appropriate for your system. For example:
    - Set Web server name = RTDWebServer
    - Set the Web server host name or IP address
    - Leave the Web server port value - that you set in step 14 - as 8080

    Click **Next**.

12. In the Web Server Definition (Part 2) screen, you can change the installation directory paths for the web server and web server plug-ins.

    Oracle recommends that you change the Web server plug-in installation directory path from its default location under HTTPServer to (under) WebSphere. By doing this, you will not have to subsequently edit the httpd.conf configuration file.



Profile Management Tool 8.0

Web Server Definition (Part 2)

Specify the remaining information about the Web server to complete the Web server definition.

Web server installation directory path: C:\IBM\HTTPServer

Web server plug-in installation directory path: C:\IBM\WebSphere\Plugins

13. Review the information in the Profile Creation Summary screen, then click **Create**.

14. When the profile is created successfully, the Profile Creation Complete screen includes an option (Launch the First steps console) to enable you to open the Customization Toolbox and run the Web Server Plug-ins Configuration Tool.

    If you want to run the Web Server Plug-ins Configuration Tool at this point, select **Launch the First steps console** and click **Finish**, else simply click **Finish**.

4.2 Configuring the Web Server Plug-in

Configuring the Web Server plug-in is to enable IBM HTTP server to work together with WebSphere Application Server. You perform this configuration using the WebSphere Customization Toolbox.

Before you configure the Web Server plug-in, ensure that the following have been installed:

- WebSphere Application Server
- WebSphere Customization Toolbox
- IBM HTTP Server
- WebSphere Plug-ins

To configure the WebServer plug-in, complete the following steps:

1. Open the WebSphere Customization Toolbox.

   If you did not do this through the Launch the First steps console option of the Profile Management tool at the end of profile creation, you can launch the Customization Toolbox by running `wct.sh` or `wct.bat` from the directory *WAS_HOME*`/C:/IBM/WebSphere/Toolbox/WCT`.

   For example:

   - `C:/IBM/WebSphere/Toolbox/WCT/wct.sh -perspective com.ibm.ws.wct.plugins.perspective`

2. In the **List of provided tools** listbox, select **Web Server Plug-ins Configuration Tool**, click **Launch Selected Tool**, then click **Add...**

3. In the Add Web Server Plug-In screen, click **Browse**. Select the same plug-in location as the Web server plug-in installation directory path selected in step 12 of Section 4.1, "Installing WebSphere Application Server and Associated Components."

   Click **Finish**, then click **Create...**

4. On the Web Server Selection screen, select **IBM HTTP Server V8**, then click **Next**.

5. If the Web Server Architecture Selection screen appears, select the architecture (64-bit or 32-bit), then click **Next**.

6. In the Web Server Configuration File Selection screen, click **Browse**.

   Specify the `httpd.conf` file of the IBM HTTP Server that you installed (typically, under *WAS_HOME*`/HTTPServer/conf`). The configuration tool will make changes to this file and include the plug-in needed by WebSphere Application Server.

   In this screen, specify also the Web server port that you set in step 14 of Section 4.1, "Installing WebSphere Application Server and Associated Components."

7. In the IBM HTTP Server Administration Server screen, perform the following:

- Select **Setup IBM HTTP Server Administration Server** *(leave the default value)*
- Keep the suggested HTTP Administration Port
- For the User ID and Password, specify the same information that you provided in step 7 of Section 4.1, "Installing WebSphere Application Server and Associated Components"
- Click **Next**.

8. In the Setup IBM HTTP Server Administration Server screen, specify the system user id and group to enable IBM HTTP Server to be started on operating system startup.

   This user and group may already exist, or you can select the option to enable the WebSphere Customization Toolbox to create them.

   Click **Next**.

9. In the Web Server Definition Name screen, specify a unique web server definition name, for example, RTDWebServer. Click **Next**.

10. In the Configuration Scenario Selection screen, you can either select a local or remote configuration scenario. Note that if you select the remote option, you must perform some post-configuration steps.

    If you select **(Remote) Host name or IP address of the application server**, then enter the host name or IP address of the WebSphere Application Server that you are configuring the web server plug-in for, and click **Next**.

    If you select **(Local) Installation of WebSphere Application Server**, then select the Application Server directory (*WAS_HOME*`/WebSphere/AppServer`), and click **Next**. Select the application server profile that you created inSection 4.1, "Installing WebSphere Application Server and Associated Components," that is, the profile for the managed node, not the deployment manager profile. Then click **Next**.

11. Review the Configuration Summary, then click **Configure**. When the configuration has completed, click **Finish**.

12. If you selected to install a remote configuration, additional steps are needed to complete the configuration. These steps are listed in the configuration roadmap:

    a. Select the **Launch the plug-in configuration roadmap** checkbox.

    b. Click **Finish**.

13. Verify the plug-in configuration by examining `httpd.conf` (in the `HTTPServer/conf` directory).

    If the port for property **Listen** is not `8080`, then change it to that value.

    If the port for property **ServerName** is not `8080`, then change it to that value.

    At the end of the file, there should be a **WebSpherePluginConfig** entry with the location of the `plugin-cfg.xml` file. For example:

    ```
    WebSpherePluginConfig /IBM/WebSphere/Plugins/config/RTDWebServer/plugin-cfg.xml
    ```

After your Web Server plug-in has been configured and is ready, you can now use the WebSphere Administrative Console to configure the rest of the elements needed for Oracle RTD deployment.

4.3 Starting WebSphere and Launching the Administrative Console

Starting the deployment manager enables access to the WebSphere Administrative Console (also referred to as WebSphere Integrated Solutions Console).

Note: In the following instructions, <*dep_mgr_prof_name*> and <*app_srvr_prof_name*> are the respective names of the Deployment Manager profile and Application server profile specified in step 5 of Section 4.1, "Installing WebSphere Application Server and Associated Components."

To start the deployment manager, run the **`startManager.sh`** or **`startManager.bat`** command, located in:

- *WAS_HOME*`/WebSphere/AppServer/profiles/`<*dep_mgr_prof_name*>`/bin`

To start the managed node, run the **`startNode.sh`** or **`startNode.bat`** command, located in:

- *WAS_HOME*`/WebSphere/AppServer/profiles/`<*app_srvr_prof_name*>`/bin`

4.3.1 Launching the WebSphere Administrative Console

Before you launch the WebSphere Administrative Console, you need to know the name of the **WC_adminhost_secure** port that appeared in step 9 of Section 4.1, "Installing WebSphere Application Server and Associated Components."

If you did not note this down, you can find this value in the properties file at:

- *WAS_HOME*`/WebSphere/AppServer/profiles/`<*dep_mgr_prof_name*>`/properties/portdef.props`

To launch the WebSphere Administrative Console, open a browser and enter a URL according to the following format:

- `https://`<*`server`*>:<*`wc_adminhost_secure_port`*>`/ibm/console`

For example:

- `https://localhost:9045/ibm/console`

4.3.2 Setting Synchronize Changes with Nodes

Synchronizing changes with nodes is not an essential step, but it makes it easier to perform the configuration steps. Otherwise you will have to synchronize nodes manually every time you save something.

Changes are saved into the deployment manager, which keeps a master copy. However, nodes are not aware of the changes until the changes are synchronized. Saving your changes does not mean they are automatically effected throughout the system, but turning this feature on ensures that synchronization commands are issued automatically, thereby keeping all your nodes in sync. After you have finished the major configuration steps needed initially for Oracle RTD, you may decide to turn this option off in your production environment.

Table 4–3 Synchronize Changes With Nodes

Access path	Object	Summary Description
System administration > **Console Preferences**	**Synchronize changes with Nodes** checkbox	Select this checkbox.

Note: The WebSphere Customization Toolbox created a server called server1. This is not required for system topologies based on servers in a cluster, and can be safely deleted.

4.4 Completing Web Server Setup

After configuring the web server in the WebSphere Customization Toolbox, you complete the web server setup in WebSphere Administrative Console.

This section contains the following topics:

- Section 4.4.1, "Generating and Propagating Web Server Plug-in"
- Section 4.4.2, "Starting the Web Server"

4.4.1 Generating and Propagating Web Server Plug-in

To generate and propagate the web server plug-in, perform the following steps in the WebSphere Administrative Console:

Table 4–4 Web Server Plug-in Generation and Propagation

Access path	Object	Summary Description
Servers > **Server Types** > **Web servers**	RTDWebServer (or the web server name set up in step 11 of Section 4.1, "Installing WebSphere Application Server and Associated Components")	1. Select the checkbox for the web server. 2. Click **Generate Plug-in**. 3. Click **Propagate Plug-in**.

4.4.2 Starting the Web Server

Table 4–5 Starting the Web Server

Access path	Object	Summary Description
Servers > **Server Types** > **Web servers**	RTDWebServer (or the web server name set up in step 11 of Section 4.1, "Installing WebSphere Application Server and Associated Components")	1. Select the checkbox for the web server. 2. Click **Start**.

4.5 Configuring WebSphere for Oracle RTD

This section contains the following topics:

- Section 4.5.1, "Setting Global Security Options"
- Section 4.5.2, "Creating Cluster and Server"
- Section 4.5.3, "Adding Virtual Host Port 8080"

- Section 4.5.4, "Setting Server Properties"
- Section 4.5.5, "Setting Server Administration Properties"
- Section 4.5.6, "Setting Server Session Management Properties"
- Section 4.5.7, "Configuring the JDBC Subsystem"
- Section 4.5.8, "Creating Groups and Users for Oracle RTD"
- Section 4.5.9, "Adding Virtual Host HTTP Address for Servers"

4.5.1 Setting Global Security Options

To set the correct global security options for Oracle RTD, perform the steps described (or confirm that the options are as described) in the following table:

Table 4–6 Creating the Cluster and Servers

Access path or Screen area	Targeted objects	Summary Description
Security > **Global security**	Settings for Administrative, Application, and Java 2 security	**1.** Select **Enable administrative security**. **2.** Select **Enable application security**. **3.** Do *not* select **Use Java 2 to restrict application access to local resources.**
Messages	Save link	Click **Save**.

Note: In all the following WebSphere Integrated Solutions Console processes, as in the one described in the preceding table, click **Save** at the end of the process.

4.5.2 Creating Cluster and Server

To create the cluster and a single server, perform the following steps:

Table 4–7 Creating the Cluster and Servers

Access path or Screen area	Targeted objects	Summary Description
Servers > **Clusters >** **WebSphere application server clusters >** **New**	Cluster + one server *(Suggested names, used throughout these sections):* - *RTDCluster* - *RTDServer1*	**1. Cluster name**: Cluster name, for example, RTDCluster **2. Member name**: Server name, for example, RTDServer1 **Select node**: Application server node name created during profile creation (see step 6 of Section 4.1, "Installing WebSphere Application Server and Associated Components") **Weight:** Keep default value *(2)* **Generate unique HTTP ports:** Keep default value *(selected)* **Select basis for first cluster member**: `Create the member using an application server template >(default)` **3. Next > Finish**

4.5.3 Adding Virtual Host Port 8080

This operation is to define the host alias to be used by cluster servers, so that the Web Server plug-in can forward requests coming from IBM HTTP Server at port 8080.

To add virtual host port 8080, perform the step shown in the following table.

Table 4–8 Adding Virtual Host Port 8080

Access path or Screen area	Targeted objects	Summary Description
Environment > **Virtual hosts**	**default_host**	**Host Aliases > New** - Host Name: ***** - Port: **8080**

4.5.4 Setting Server Properties

To set the server JVM properties, perform the following steps:

1. In the Integrated Solutions Console, expand **Servers**, then **Server Types** > **WebSphere application servers**.
2. Select the server, for example, RTDServer1.
3. Select **Java and Process Management > Process Definition > Java Virtual Machine > Custom Properties > New**.

   At this point, you must enter certain mandatory properties, and you can enter other optional properties.

   The mandatory properties are as follows:

   - **rtd.instanceName** - the name of the node, can be any name.

     The rtd.instanceName value can only contain alphanumerics, underscore, dash, and period.

   - **rtd.HTTPHostAndPort** - in format <host>:<port> - specifies the RPC host:port for this node.

     Set the port to value of *WC_defaulthost* displayed at the path *Communications* > *Ports* for this server.

     Important: Each node in an Oracle RTD cluster must be assigned these two JVM system properties, rtd.instanceName and rtd.HTTPHostAndPort, which must be unique to the node across the cluster.

   - **org.eclipse.emf.ecore.EPackage.Registry.INSTANCE**

     This must be set to **com.sigmadynamics.emf.util.SDEMFRegistry**.

Notes: **1.** The system property, **rtd.skipIlsTestLoad**, can be used to bypass test loading of an Inline Service (ILS) during ILS deployment. If unspecified in the JVM start parameters, the property has a default value of false, which means ILS test loading will be performed during ILS deployment.

If the logging level is set to DEBUG, a message similar to the following can be seen in the server log file:

```
2013-12-10 11:04:10,142 DEBUG [DeployAppCommand] Performing
test load on CrossSell with deployment state 5.
```

Set the property to true, that is, add **-Drtd.skipIlsTestLoad=true** to bypass ILS test loading. This may be helpful in ILS redeployments from Decision Center in a clustered environment.

2. If your Inline Services have many external rules, add the following JVM parameter to improve the initial load time of these external rules:

-Dsun.reflect.inflationThreshold=2147483647

4.5.5 Setting Server Administration Properties

To set the server administration properties, perform the steps in the following table.

Table 4–9 Setting Server Administration Properties

Access path or Screen area	Targeted objects	Summary Description
Servers > **Server Types** > **WebSphere application servers**	**RTDServer1**	Select the server, then add the server administration property in the next step.

Administration > Administration services > Custom properties > New	(See Summary Description)	Name=`com.ibm.websphere.mbeans.disableRouting`
		Value=`<on>OracleRTD:*</on>`

4.5.6 Setting Server Session Management Properties

To set the server session management properties, perform the following steps:

Table 4–10 Setting Server Session Management Properties

Access path or Screen area	Targeted objects	Summary Description
Servers > **Server Types** > **WebSphere application servers**	**RTDServer1**	Select the server, then add the server session management property in the next step.

Session Management > Custom properties > New	(See Summary Description)	Name=`InvalidateOnUnauthorizedSessionRequestException`
		Value=`true`

4.5.7 Configuring the JDBC Subsystem

Configuring the JDBC subsystem in WebSphere consists of the following three stages:

- Creating a JDBC provider - this specifies which database and database driver to use
- Creating a J2C Authentication Alias - this specifies how to login to your database
- Creating a JDBC data source - this enables Oracle RTD to find and access the database

This section contains the following topics:

- Considerations and Prerequisite
- Creating JDBC Provider
- Creating J2C Authentication Alias
- Creating Data Source
- Restarting the Node
- Testing Data Source Connection

4.5.7.1 Considerations and Prerequisite

For JDBC, the concept of scope is important. All JDBC elements set up for Oracle RTD should be defined in cluster scope, which localizes them to the cluster, and hides them from other software components that you may install or may have installed outside of the cluster.

Important: As a prerequisite, you must perform the instructions in *RTD_HOME*`\lib\jdbc\readme.txt` to download and install the JDBC driver `ojdbc6.jar`.

The directory into which to put the driver must be accessible to the WebSphere Application Server.

For example, create a directory `oracleJDBC` under *WAS_HOME*`/Websphere/AppServer/` and place the .jar file there.

You can optionally create a cluster-scoped WebSphere variable, such as ORACLE_JDBC_DRIVER_PATH, to point to this directory.

In a subsequent setup step, you will need to specify this directory - either explicitly or using the WebSphere variable - in the Class path for the JDBC provider.

4.5.7.2 Creating JDBC Provider

To create a JDBC provider, perform the following steps:

Table 4–11 Creating JDBC Provider

Access path or Screen area	Targeted objects	Summary Description
Resources > **JDBC** > **JDBC providers >** **New**	JDBC provider	**1.** Scope: Select cluster scope **Database type**: Oracle **Provider type**: Oracle JDBC Driver **Implementation type**: Connection pool data source **Name**: Keep the default *(Oracle JDBC Driver)* Click **Next**. **2.** **Class path**: Specify the fully-qualified name for the ojdbc6.jar file in the directory in which you placed it. For example, if you previously created the cluster-scoped WebSphere variable ORACLE_JDBC_DRIVER_PATH, enter ${ORACLE_JDBC_DRIVER_PATH}/ojdc6.jar **3.** **Next > Finish**

4.5.7.3 Creating J2C Authentication Alias

To create a J2C authentication alias, perform the following steps:

> **Note:** Unless you must have compatibility with earlier releases, you do not need to select the option **Prefix new alias names with the node name of the cell**.

Table 4–12 Creating J2C Authentication Alias

Access path or Screen area	Targeted objects	Summary Description
Security > **Global security >** **Java Authentication and Authorization Service** > **J2C authentication data >** **New**	J2C authentication alias	Enter the Alias name *(suggested name RTDDS_Auth)*. Enter UserId and Password, for the database runtime user.

4.5.7.4 Creating Data Source

To create the data source for the Oracle RTD database, perform the following steps:

Table 4–13 Creating Data Source

Access path or Screen area	Targeted objects	Summary Description
Resources > **JDBC** > **Data sources** > **New**	Data source for SDDS	1. Scope: Select cluster scope **Data source name** can be any name **JNDI name** must be SDDS Click **Next**. 2. **Select an existing JDBC provider**: Select the JDBC provider that you created in Section 4.5.7.2, "Creating JDBC Provider." Click **Next**. 3. **URL:** Enter the URL using the format: *jdbc:oracle:thin:@{database_host}:{database_port}:{database_name}* For example, jdbc:oracle:thin:@myhost:1521:orcl **Data store helper class name:** Oracle11g data store helper Click **Next**. 4. **Component-managed authentication alias**: Select the alias that you created in Section 4.5.7.3, "Creating J2C Authentication Alias." Keep the default settings of None for **Mapping-configuration alias** and **Container-manager authentication alias**. **Next** > **Finish**.

4.5.7.5 Restarting the Node

Before you can use - or even test - your new data source, you have to restart the WebSphere managed node. You do not need to restart the Deployment Manager.

To restart the node, perform the following steps:

1. Stop the node from either from the console (System Administration -> Nodes) or by using the appropriate command file from the command line, using the following format:

   WAS_HOME/WebSphere/AppServer/profiles/<*app_srvr_prof_name*>/bin/<*stopnode_cmd*> -username <*adminuser*> -password <*adminpass*>

   where:

   - <*app_srvr_prof_name*> is the name of the Application server profile specified in step 5 of Section 4.1, "Installing WebSphere Application Server and Associated Components."
   - <*stopnode_cmd*> is either **stopNode.sh** for Linux or **stopNode.bat** for Windows

     of the Application server profile specified in step 5 of Section 4.1, "Installing WebSphere Application Server and Associated Components.".
   - <*adminuser*> and <*adminpass*> are administrative user name and password respectively.

2. Start the node by using the appropriate command file from the command line, using the following format *(note that username and password are not required)*:

   WAS_HOME/WebSphere/AppServer/profiles/<*app_srvr_prof_name*>/bin/<*startnode_cmd*>

   where:

- <*app_srvr_prof_name*> is the name of the Application server profile specified in step 5 of Section 4.1, "Installing WebSphere Application Server and Associated Components."
- <*startnode_cmd*> is either **startNode.sh** for Linux or **startNode.bat** for Windows

4.5.7.6 Testing Data Source Connection

To test the data source connection, perform the following steps:

Table 4–14 Testing Data Source Connection

Access path or Screen area	Targeted objects	Summary Description
Resources > **JDBC** > **Data sources**	Data source for SDDS database	Select the data source, then **Test connection**.

4.5.8 Creating Groups and Users for Oracle RTD

This section consists of the following topics:

- Section 4.5.8.1, "Creating Groups"
- Section 4.5.8.2, "Creating Users for Oracle RTD"

Note: After creating or modifying user groups and/or users you must restart the WebSphere managed node, as described in Section 4.5.7.5, "Restarting the Node".

4.5.8.1 Creating Groups

To create the groups for Oracle RTD and to make each of the groups a member of RTDUserGroup, perform the following steps:

Table 4–15 Creating Groups for Oracle RTD

Access path or Screen area	Targeted objects	Summary Description
Users and Groups > Manage Groups > Create...	Standard Oracle RTD groups: ■ RTDAdminGroup ■ RTDDCEditorGroup ■ RTDDCUserGroup ■ RTDStudioDeployerGroup ■ RTDStudioDownloaderGroup ■ RTDBatchAdminGroup ■ RTDChoiceEditorGroup ■ RTDUserGroup	Create the groups ■ Enter the group name, then **Create**, then **Close**. ■ Repeat for all the listed standard Oracle RTD groups.
Users and Groups > Manage Groups	(See Summary Description)	Add users to the groups ■ For each group: **Add Users**, **Search**, select users to add, then **Add**, then **Close**
Users and Groups > Manage Groups	RTDUserGroup	Add the other standard Oracle RTD groups as members of RTDUserGroup ■ *For RTDUserGroup:* **Members** tab> **Add Groups** > select the other standard Oracle RTD groups, then **Close**

4.5.8.2 Creating Users for Oracle RTD

To create users, perform the following step:

Table 4–16 Creating Users for Oracle RTD

Access path or Screen area	Targeted objects	Summary Description
Users and Groups > Manage Users > Create...	Users (including administrative user)	Enter the administrative user name *(for example, rtdadmin)* and password, then **Create**, then **Close**. Repeat for other users.

You can then assign users to one or more groups, for example:

- User rtduser to group RTDDCUserGroup
- User rtdadmin to group RTDAdminGroup
- User rtddesigner to groups RTDChoiceEditorGroup, RTDDCEditorGroup, RTDStudioDeployerGroup, RTDStudioDownloaderGroup

4.5.9 Adding Virtual Host HTTP Address for Servers

This process consists of getting HTTP addresses for each server, then creating an HTTP virtual port for each unique server port.

4.5.9.1 Getting HTTP Addresses For Each Server

To get HTTP addresses for each server, perform the following steps:

Table 4–17 Getting HTTP Addresses For Each Server

Access path or Screen area	Targeted objects	Summary Description
Servers > **Server Types** > **WebSphere application servers**	**RTDServer1**	Select the server, then Ports. Write down the *`WC_defaulthost`* for the server. This is used to connect with this server via HTTP.

4.5.9.2 Creating an HTTP Virtual Port For Each Unique Server Port

To create an HTTP virtual port for each unique server port, perform the following steps:

Table 4–18 Creating HTTP Virtual Ports for Each Server Port

Access path or Screen area	Targeted objects	Summary Description
Environment > **Virtual hosts**	Host Aliases for default_host for each server	Select default_host, then Host Aliases > New. Create, for each server: ■ **Host Name** = * ■ **Port** = the WC_defaulthost port noted from Section 4.5.9.1, "Getting HTTP Addresses For Each Server" Note that you do not have to add the same port a second time for different machines.

Note: Oracle recommends that, at this point, you do *not* start the server or cluster. This is to enable you to make configuration changes during the deployment of Oracle RTD, and thereby avoid the need to restart the server or cluster after the deployment.

4.6 Deploying RTD.ear to the Cluster

This process consists of deploying `RTD.ear` to the cluster, then starting the OracleRTD application.

This section contains the following topics:

- Section 4.6.1, "Deploying RTD.ear to Cluster (Fast Path option)"
- Section 4.6.2, "Deploying RTD.ear to Cluster (Detailed option)"
- Section 4.6.3, "Changing Class Loader Priority"
- Section 4.6.4, "Starting the Server"

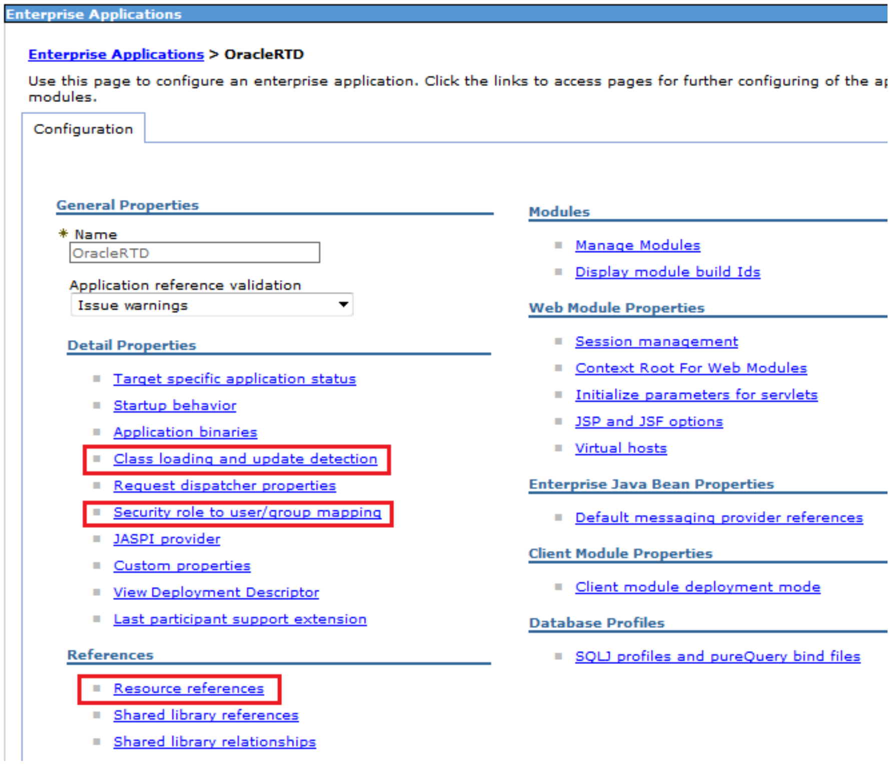
4.6.1 Deploying RTD.ear to Cluster (Fast Path option)

To deploy `RTD.ear` to the cluster using the Fast Path option, perform the following steps:

Table 4–19 WebLogic Installation Screens

Access path or Screen area	Targeted objects	Summary Description
Applications > **New Application** > **New Enterprise Application**	RTD.ear	**Local file system** or **Remote file system**, click **Browse...** and browse to *`RTD_HOME`*`/package/RTD.ear`. **1.** **Fast path**, then click**Browse...** and browse to *`RTD_HOME`*`/package/RTD.ear`. **2.** **Application name**: OracleRTD **3.** Deploy to the cluster (and if webserver set up, to the webserver).

Following the successful deployment of RTD.ear using the Fast Path option, you must perform three post-deployment processes, as highlighted in the General Properties list for OracleRTD:



Enterprise Applications > OracleRTD

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the ap modules.

Name: OracleRTD

Application reference validation: Issue warnings

Enterprise Applications
Configuration
General Properties
Detail Properties
Target specific application status
Startup behavior
Application binaries
Class loading and update detection
Request dispatcher properties
Security role to user/group mapping
JASPI provider
Custom properties
View Deployment Descriptor
Last participant support extension
References
Resource references
Shared library references
Shared library relationships
Modules
Manage Modules
Display module build Ids
Web Module Properties
Session management
Context Root For Web Modules
Initialize parameters for servlets
JSP and JSF options
Virtual hosts
Enterprise Java Bean Properties
Default messaging provider references
Client Module Properties
Client module deployment mode
Database Profiles
SQLJ profiles and pureQuery bind files

The **Class loading and update detection** process is described in Section 4.6.3, "Changing Class Loader Priority."

The **Security role to user/group mapping** process is described in Section 4.7.2, "Mapping Roles to Groups After RTD.ear Deployment."

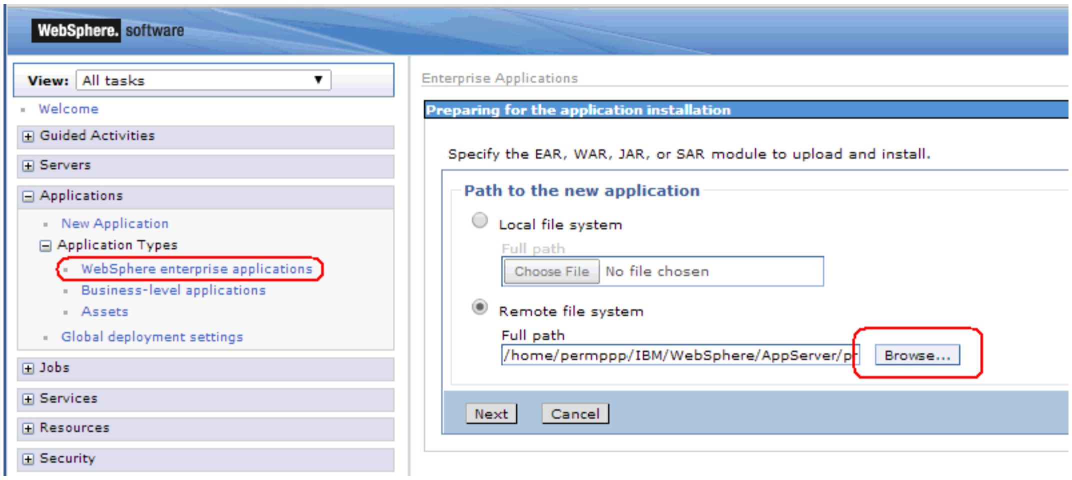
When you click **Resource references**, the **Resource references** screen that appears is the same as the **Map resource references to resources** screen that appears in step 6 of Section 4.6.2, "Deploying RTD.ear to Cluster (Detailed option)" - which describes Step 7 of the Detailed deployment process.

You must perform the same task as described in that section, that is, you must ensure that all resource references to SDDS are matched to the data source target resource with JNDI name SDDS.

4.6.2 Deploying RTD.ear to Cluster (Detailed option)

To deploy RTD.ear to the cluster using the Detailed option, perform the following steps:.

1. Select **WebSphere enterprise applications**, click **Install**, select **Remote file system**, then select RTD.ear from where you unpacked it.



Specify the EAR, WAR, JAR, or SAR module to upload and install.

○ Local file system

Full path

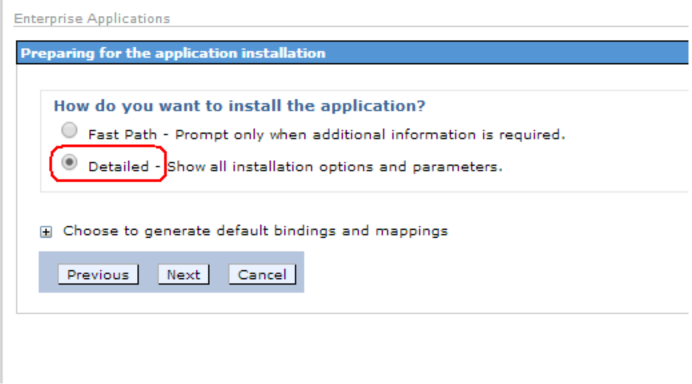
No file chosen

● Remote file system

Full path

/home/permppp/IBM/WebSphere/AppServer/p

2. Select Detailed.

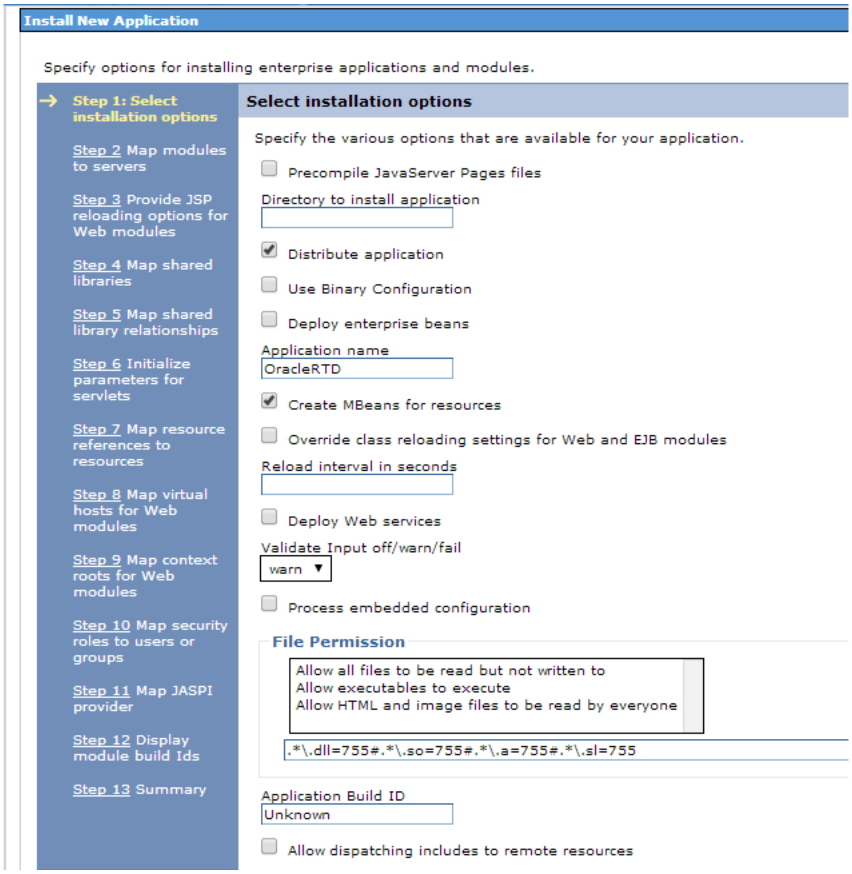

Enterprise Applications
Preparing for the application installation
Choose to generate default bindings and mappings
Previous
Next
Cancel

How do you want to install the application?

○ Fast Path - Prompt only when additional information is required.

● Detailed - Show all installation options and parameters.

3. Leave all defaults in deployment Step 1, ensuring that the Application name is OracleRTD.



Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options
Step 2 Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map shared library relationships
Step 6 Initialize parameters for servlets
Step 7 Map resource references to resources
Step 8 Map virtual hosts for Web modules
Step 9 Map context roots for Web modules
Step 10 Map security roles to users or groups
Step 11 Map JASPI provider
Step 12 Display module build Ids
Step 13 Summary

Select installation options

Specify the various options that are available for your application.

☐ Precompile JavaServer Pages files

Directory to install application

☑ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name: OracleRTD

☑ Create MBeans for resources

☐ Override class reloading settings for Web and EJB modules

Reload interval in seconds

☐ Deploy Web services

Validate Input off/warn/fail: warn

☐ Process embedded configuration

File Permission

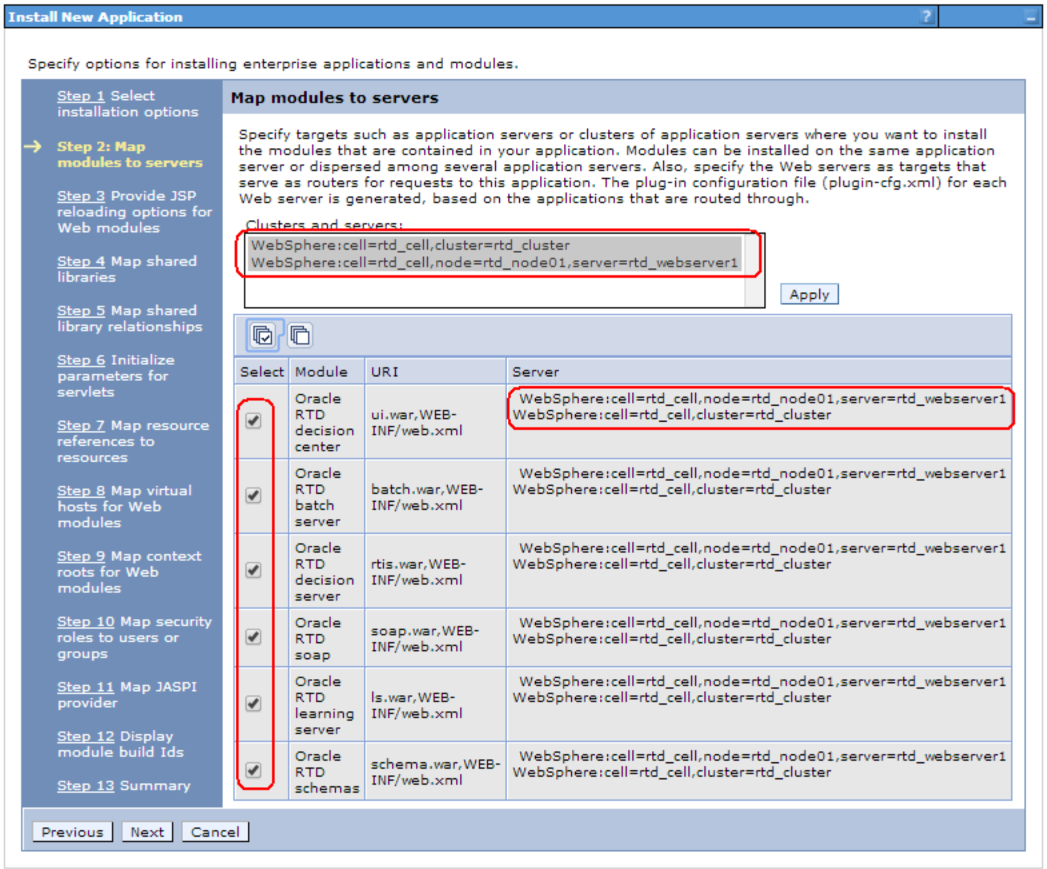
Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

Application Build ID: Unknown

☐ Allow dispatching includes to remote resources

4. In the **Map modules to servers** screen (Step 2), ensure that you target the deployment to your cluster and web server rtd_cluster and rtd_webserver1 in the example screenshot.


Install New Application
Step 1 Select installation options
Step 2: Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map shared library relationships
Step 6 Initialize parameters for servlets
Step 7 Map resource references to resources
Step 8 Map virtual hosts for Web modules
Step 9 Map context roots for Web modules
Step 10 Map security roles to users or groups
Step 11 Map JASPI provider
Step 12 Display module build Ids
Step 13 Summary
Apply
Previous
Next
Cancel

Specify options for installing enterprise applications and modules.

Map modules to servers

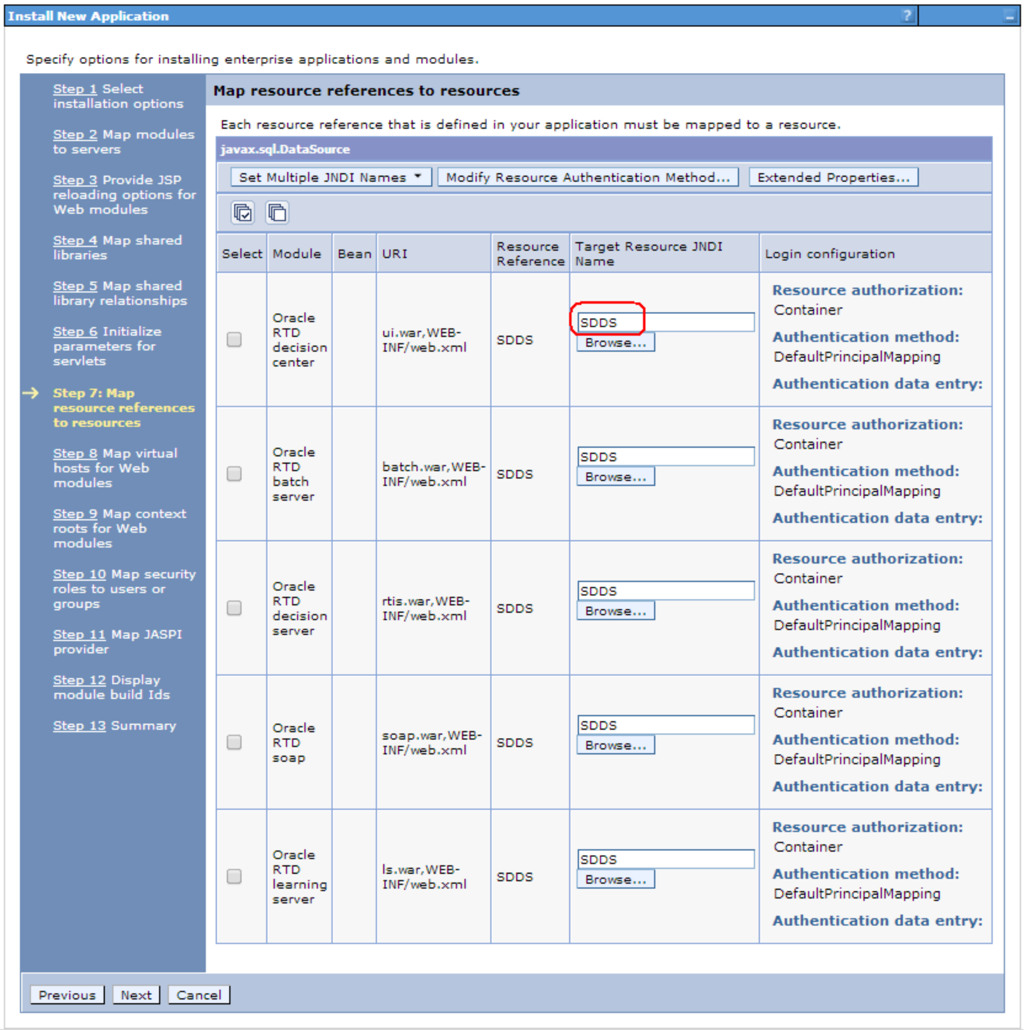
Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

WebSphere:cell=rtd_cell,cluster=rtd_cluster
WebSphere:cell=rtd_cell,node=rtd_node01,server=rtd_webserver1

Select	Module	URI	Server
☑	Oracle RTD decision center	ui.war,WEB-INF/web.xml	WebSphere:cell=rtd_cell,node=rtd_node01,server=rtd_webserver1 WebSphere:cell=rtd_cell,cluster=rtd_cluster
☑	Oracle RTD batch server	batch.war,WEB-INF/web.xml	WebSphere:cell=rtd_cell,node=rtd_node01,server=rtd_webserver1 WebSphere:cell=rtd_cell,cluster=rtd_cluster
☑	Oracle RTD decision server	rtis.war,WEB-INF/web.xml	WebSphere:cell=rtd_cell,node=rtd_node01,server=rtd_webserver1 WebSphere:cell=rtd_cell,cluster=rtd_cluster
☑	Oracle RTD soap	soap.war,WEB-INF/web.xml	WebSphere:cell=rtd_cell,node=rtd_node01,server=rtd_webserver1 WebSphere:cell=rtd_cell,cluster=rtd_cluster
☑	Oracle RTD learning server	ls.war,WEB-INF/web.xml	WebSphere:cell=rtd_cell,node=rtd_node01,server=rtd_webserver1 WebSphere:cell=rtd_cell,cluster=rtd_cluster
☑	Oracle RTD schemas	schema.war,WEB-INF/web.xml	WebSphere:cell=rtd_cell,node=rtd_node01,server=rtd_webserver1 WebSphere:cell=rtd_cell,cluster=rtd_cluster

5. In deployment Steps 3 to 6, accept the defaults.
6. In the **Map resource references to resources** screen (Step 7), ensure that the deployer picks up the JDBC data source with JNDI name SDDS.


Install New Application
Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map shared library relationships
Step 6 Initialize parameters for servlets
Step 7: Map resource references to resources
Step 8 Map virtual hosts for Web modules
Step 9 Map context roots for Web modules
Step 10 Map security roles to users or groups
Step 11 Map JASPI provider
Step 12 Display module build Ids
Step 13 Summary
javax.sql.DataSource
Set Multiple JNDI Names
Modify Resource Authentication Method...
Extended Properties...
Previous
Next
Cancel

Specify options for installing enterprise applications and modules.

Map resource references to resources

Each resource reference that is defined in your application must be mapped to a resource.

Select	Module	Bean	URI	Resource Reference	Target Resource JNDI Name	Login configuration
☐	Oracle RTD decision center		ui.war,WEB-INF/web.xml	SDDS	SDDS Browse...	**Resource authorization:** Container **Authentication method:** DefaultPrincipalMapping **Authentication data entry:**
☐	Oracle RTD batch server		batch.war,WEB-INF/web.xml	SDDS	SDDS Browse...	**Resource authorization:** Container **Authentication method:** DefaultPrincipalMapping **Authentication data entry:**
☐	Oracle RTD decision server		rtis.war,WEB-INF/web.xml	SDDS	SDDS Browse...	**Resource authorization:** Container **Authentication method:** DefaultPrincipalMapping **Authentication data entry:**
☐	Oracle RTD soap		soap.war,WEB-INF/web.xml	SDDS	SDDS Browse...	**Resource authorization:** Container **Authentication method:** DefaultPrincipalMapping **Authentication data entry:**
☐	Oracle RTD learning server		ls.war,WEB-INF/web.xml	SDDS	SDDS Browse...	**Resource authorization:** Container **Authentication method:** DefaultPrincipalMapping **Authentication data entry:**

If it does not, click Browse… and select SDDS for all Oracle RTD modules.

7. In the **Map virtual hosts for Web modules** screen, (Step 8), select default host.
8. In the **Map context roots for Web modules** screen, (Step 9), accept the defaults.
9. In the **Map security roles to users or groups** screen (Step 10) - you can perform a number of mapping operations or keep the default mappings.

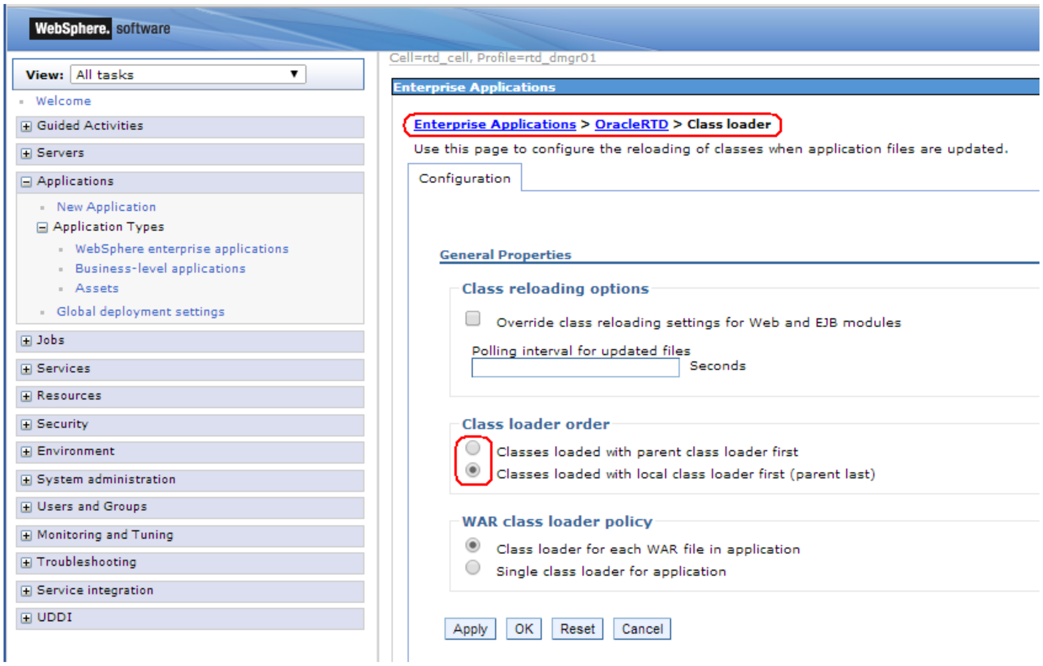
   If you are not satisfied with default mapping between Oracle RTD Roles and Websphere user groups, you can change the mapping. For instructions, including all the options, see Section 4.7, "Mapping Oracle RTD Roles to WebSphere Groups".
10. Accept defaults for the rest of deployment steps, review the summary, finish the deployment and save the configuration.

4.6.3 Changing Class Loader Priority

Note: You must perform this post-deployment step whether you deployed Oracle RTD using the Fast Path or the Detailed method.

To change the class loader priority, perform the following step:

1. For the OracleRTD application, select the property **Class loading and update detection**.
2. Select **Classes loaded with local loader class first (parent last).**



WebSphere. software

View: All tasks

- Welcome
- ⊞ Guided Activities
- ⊞ Servers
- ⊟ Applications
  - New Application
  - ⊟ Application Types
    - WebSphere enterprise applications
    - Business-level applications
    - Assets
  - Global deployment settings
- ⊞ Jobs
- ⊞ Services
- ⊞ Resources
- ⊞ Security
- ⊞ Environment
- ⊞ System administration
- ⊞ Users and Groups
- ⊞ Monitoring and Tuning
- ⊞ Troubleshooting
- ⊞ Service integration
- ⊞ UDDI

Cell=rtd_cell, Profile=rtd_dmgr01

Enterprise Applications

Enterprise Applications > OracleRTD > Class loader

Use this page to configure the reloading of classes when application files are updated.

Configuration

General Properties

Class reloading options

☐ Override class reloading settings for Web and EJB modules

Polling interval for updated files

Seconds

Class loader order

○ Classes loaded with parent class loader first

● Classes loaded with local class loader first (parent last)

WAR class loader policy

● Class loader for each WAR file in application

○ Single class loader for application

Apply | OK | Reset | Cancel

3. Click **OK**.

4.6.4 Starting the Server

At this point, the deployment manager and managed node are running, the server *(rtd_server1)* and webserver *(rtd_webserver1)* are stopped, and OracleRTD has been deployed:

Start the server, then check its log:

`{WAS_HOME}/WebSphere/AppServer/profiles/rtd_node01/logs/rtd_server1/SystemOut.log`.

There should be no exceptions or anything suspicious.

OracleRTD should have started automatically. If this has not happened, then start OracleRTD.

Now perform the verification tasks specified in Section 3.2, "Verification Tasks."

4.7 Mapping Oracle RTD Roles to WebSphere Groups

Standard Oracle RTD roles are defined in the RTD.ear file, but can be re-mapped to different WebSphere groups either during or after Oracle RTD deployment.

Table 4–20 displays the list of the standard Oracle RTD roles and the WebSphere groups (the creation of which is described in Section 4.5.8, "Creating Groups and Users for Oracle RTD") to which, by default, the roles must be mapped.

Table 4–20 Standard Oracle RTD Roles and Group Associations

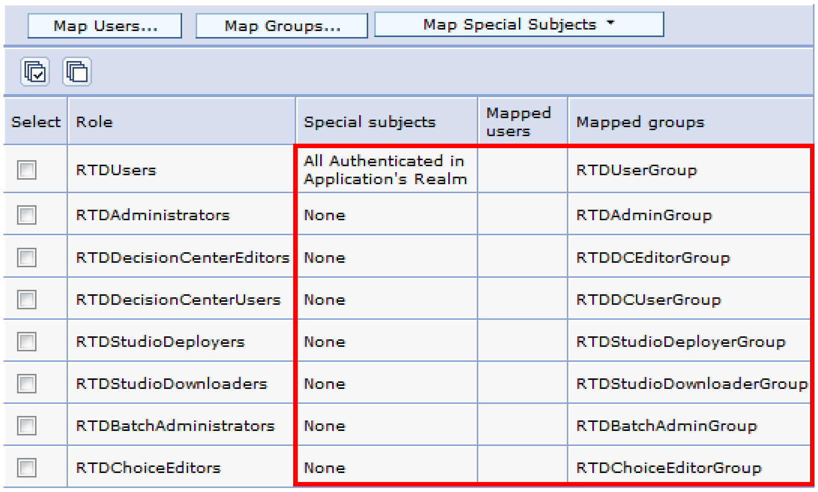
Role	Group
RTDUsers	RTDUserGroup
RTDAdministrators	RTDAdminGroup
RTDDecisionCenterEditors	RTDDCEditorGroup
RTDDecisionCenterUsers	RTDDCUserGroup
RTDStudioDeployers	RTDStudioDeployerGroup
RTDStudioDownloaders	RTDStudioDownloaderGroup
RTDBatchAdministrators	RTDBatchAdminGroup
RTDChoiceEditors	RTDChoiceEditorGroup

> **Note:** Section 5.2, "Standard Oracle RTD Roles" and its component subsections describe how default permissions are already assigned to the standard Oracle RTD roles. These become active immediately after Oracle RTD is installed and started on WebSphere.

The process of mapping the roles to groups can either be performed during the deployment of RTD.ear, or as a post-deployment process.

During the deployment, this process occurs at Step 10 of the Detailed deployment, where the screen **Map security roles to users or groups** appears.

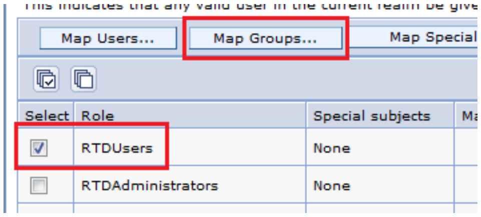
4.7.1 Changing of Roles to Groups Mapping



Select	Role	Special subjects	Mapped users	Mapped groups
☐	RTDUsers	All Authenticated in Application's Realm		RTDUserGroup
☐	RTDAdministrators	None		RTDAdminGroup
☐	RTDDecisionCenterEditors	None		RTDDCEditorGroup
☐	RTDDecisionCenterUsers	None		RTDDCUserGroup
☐	RTDStudioDeployers	None		RTDStudioDeployerGroup
☐	RTDStudioDownloaders	None		RTDStudioDownloaderGroup
☐	RTDBatchAdministrators	None		RTDBatchAdminGroup
☐	RTDChoiceEditors	None		RTDChoiceEditorGroup

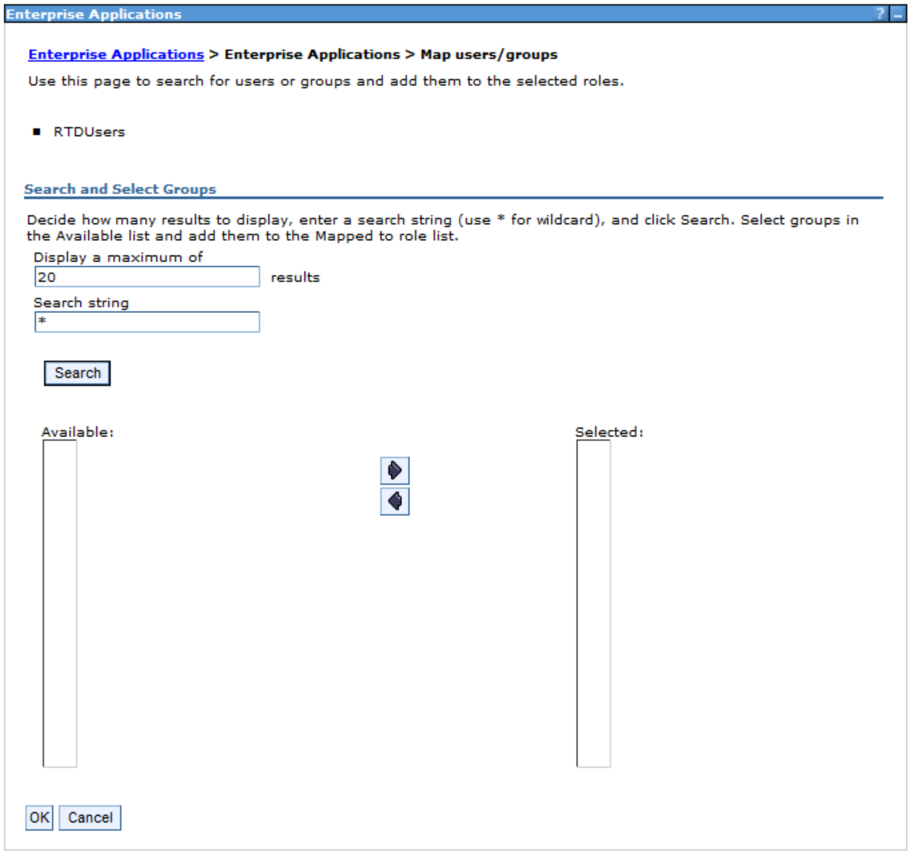
To map the standard Oracle RTD roles to the corresponding WebSphere groups, perform the following steps:

1. Select one of the roles (RTDUsers for example) and click **Map Groups**.



Select	Role	Special subjects
☑	RTDUsers	None
☐	RTDAdministrators	None

The Search and Select Groups screen appears.



Enterprise Applications > Enterprise Applications > Map users/groups

Use this page to search for users or groups and add them to the selected roles.

- RTDUsers

Search and Select Groups

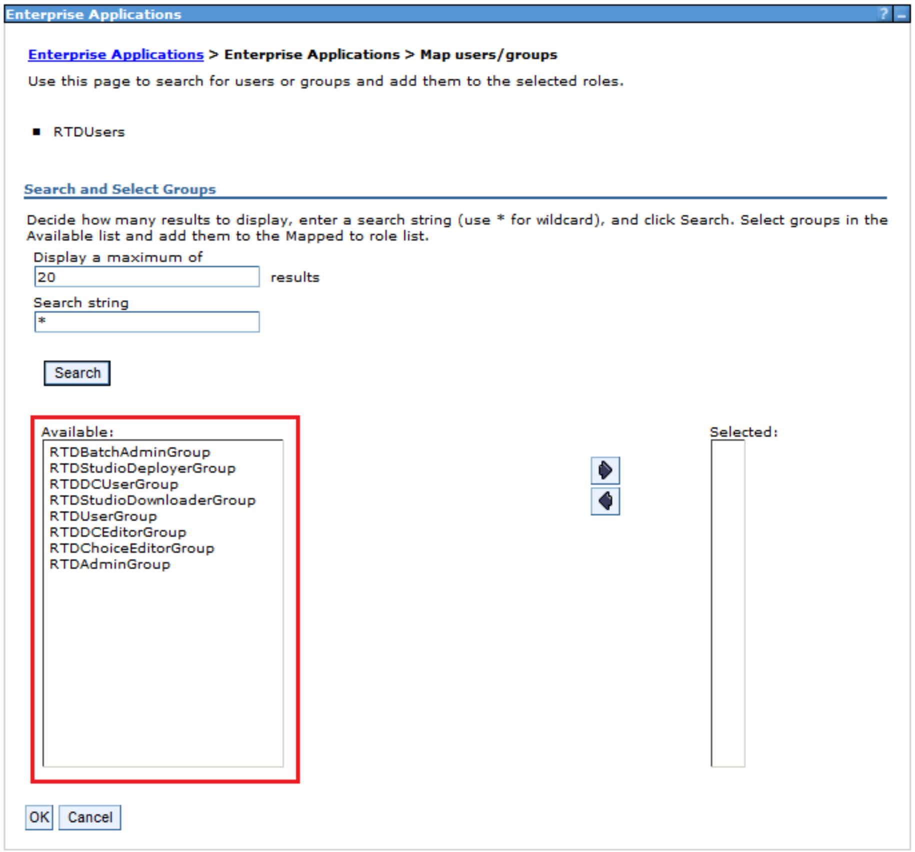
Decide how many results to display, enter a search string (use * for wildcard), and click Search. Select groups in the Available list and add them to the Mapped to role list.

Display a maximum of: 20 results

Search string: *

2. Click Search.

   This displays the list of Available user groups:



Enterprise Applications > Enterprise Applications > Map users/groups

Use this page to search for users or groups and add them to the selected roles.

- RTDUsers

Search and Select Groups

Decide how many results to display, enter a search string (use * for wildcard), and click Search. Select groups in the Available list and add them to the Mapped to role list.

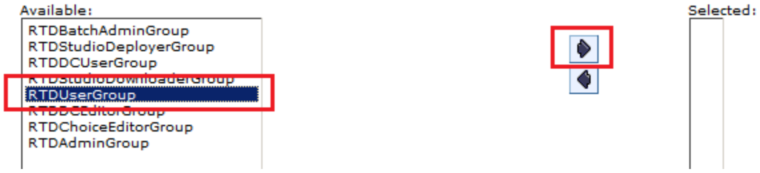
Display a maximum of 20 results

Search string *

Available:
RTDBatchAdminGroup
RTDStudioDeployerGroup
RTDDCUserGroup
RTDStudioDownloaderGroup
RTDUserGroup
RTDDCEditorGroup
RTDChoiceEditorGroup
RTDAdminGroup

Selected:

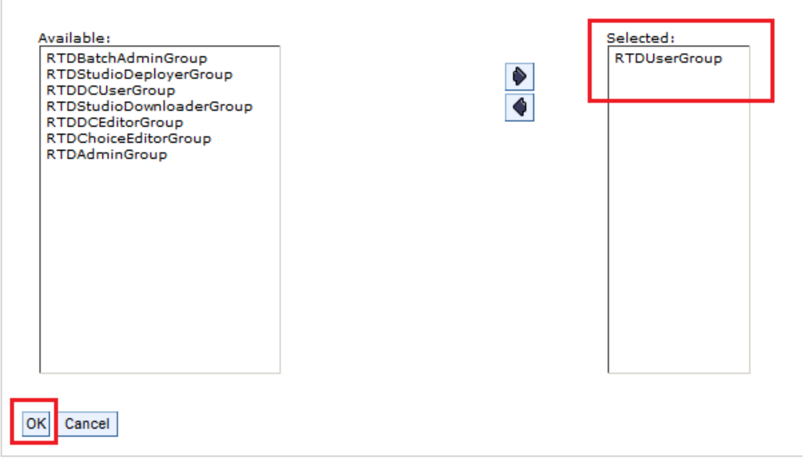
3. Select one group and move it to the Selected list.



Available:
RTDBatchAdminGroup
RTDStudioDeployerGroup
RTDDCUserGroup
RTDStudioDownloaderGroup
RTDUserGroup
RTDDCEditorGroup
RTDChoiceEditorGroup
RTDAdminGroup

Selected:

This group then appears in the Selected list.

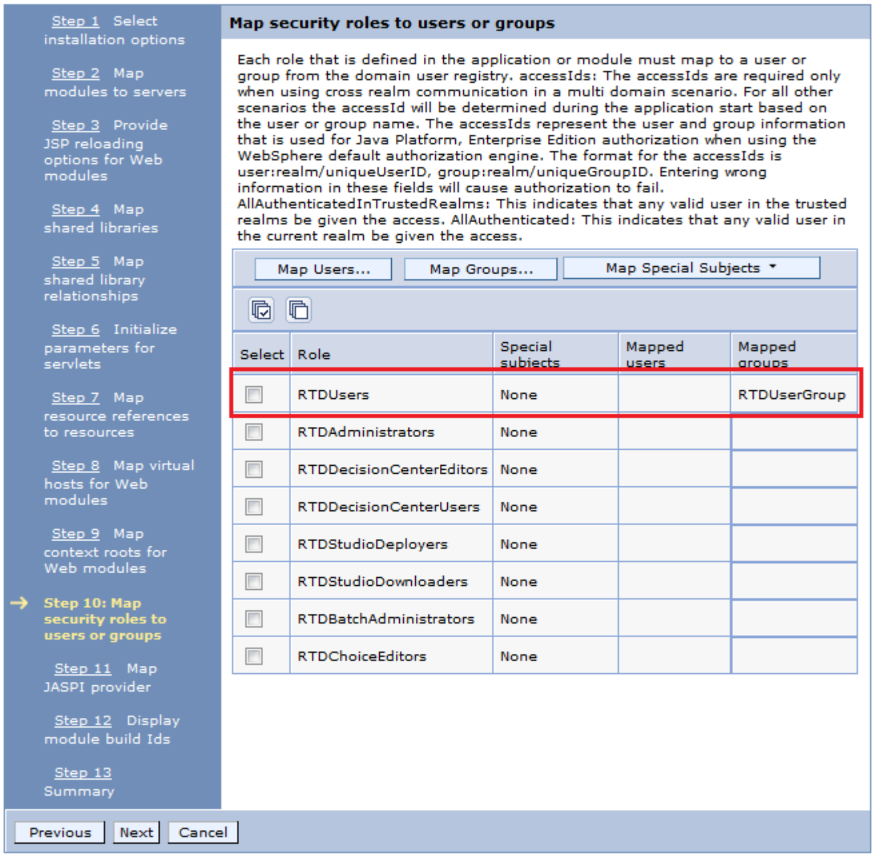


Available:
RTDBatchAdminGroup
RTDStudioDeployerGroup
RTDDCUserGroup
RTDStudioDownloaderGroup
RTDDCEditorGroup
RTDChoiceEditorGroup
RTDAdminGroup

Selected:
RTDUserGroup

4. Click **OK**.

You have successfully mapped the standard Oracle RTD role RTDUsers to the WebSphere group RTDUserGroup.

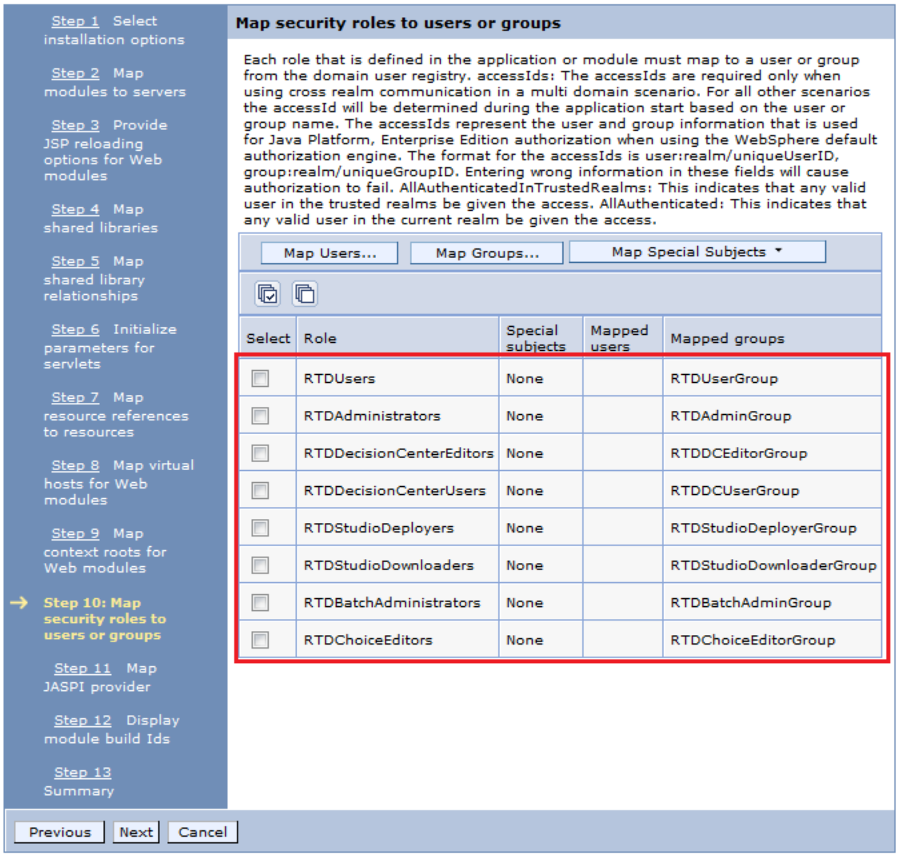

Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map shared library relationships
Step 6 Initialize parameters for servlets
Step 7 Map resource references to resources
Step 8 Map virtual hosts for Web modules
Step 9 Map context roots for Web modules
→ Step 10: Map security roles to users or groups
Step 11 Map JASPI provider
Step 12 Display module build Ids
Step 13 Summary
Map Users...
Map Groups...
Map Special Subjects
Previous
Next
Cancel

Map security roles to users or groups

Each role that is defined in the application or module must map to a user or group from the domain user registry. accessIds: The accessIds are required only when using cross realm communication in a multi domain scenario. For all other scenarios the accessId will be determined during the application start based on the user or group name. The accessIds represent the user and group information that is used for Java Platform, Enterprise Edition authorization when using the WebSphere default authorization engine. The format for the accessIds is user:realm/uniqueUserID, group:realm/uniqueGroupID. Entering wrong information in these fields will cause authorization to fail. AllAuthenticatedInTrustedRealms: This indicates that any valid user in the trusted realms be given the access. AllAuthenticated: This indicates that any valid user in the current realm be given the access.

Select	Role	Special subjects	Mapped users	Mapped groups
☐	RTDUsers	None		RTDUserGroup
☐	RTDAdministrators	None		
☐	RTDDecisionCenterEditors	None		
☐	RTDDecisionCenterUsers	None		
☐	RTDStudioDeployers	None		
☐	RTDStudioDownloaders	None		
☐	RTDBatchAdministrators	None		
☐	RTDChoiceEditors	None		

5. Now repeat steps 1 to 4 for the rest of the roles until you have everything mapped.



Step 1 Select installation options
Step 2 Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map shared library relationships
Step 6 Initialize parameters for servlets
Step 7 Map resource references to resources
Step 8 Map virtual hosts for Web modules
Step 9 Map context roots for Web modules
→ Step 10: Map security roles to users or groups
Step 11 Map JASPI provider
Step 12 Display module build Ids
Step 13 Summary

Map security roles to users or groups

Each role that is defined in the application or module must map to a user or group from the domain user registry. accessIds: The accessIds are required only when using cross realm communication in a multi domain scenario. For all other scenarios the accessId will be determined during the application start based on the user or group name. The accessIds represent the user and group information that is used for Java Platform, Enterprise Edition authorization when using the WebSphere default authorization engine. The format for the accessIds is user:realm/uniqueUserID, group:realm/uniqueGroupID. Entering wrong information in these fields will cause authorization to fail. AllAuthenticatedInTrustedRealms: This indicates that any valid user in the trusted realms be given the access. AllAuthenticated: This indicates that any valid user in the current realm be given the access.

Select	Role	Special subjects	Mapped users	Mapped groups
☐	RTDUsers	None		RTDUserGroup
☐	RTDAdministrators	None		RTDAdminGroup
☐	RTDDecisionCenterEditors	None		RTDDCEditorGroup
☐	RTDDecisionCenterUsers	None		RTDDCUserGroup
☐	RTDStudioDeployers	None		RTDStudioDeployerGroup
☐	RTDStudioDownloaders	None		RTDStudioDownloaderGroup
☐	RTDBatchAdministrators	None		RTDBatchAdminGroup
☐	RTDChoiceEditors	None		RTDChoiceEditorGroup

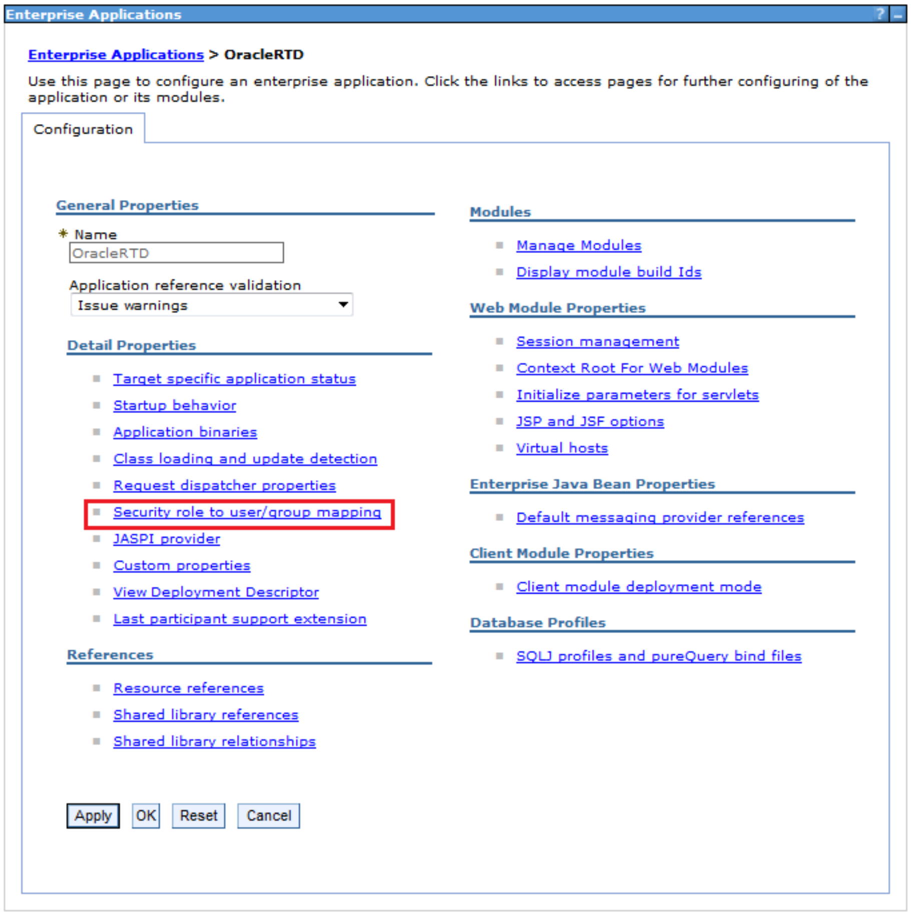
6. If you have performed these steps during—and not after—the deployment of RTD.ear, you can click **Next** and complete the remaining deployment steps, that is, continue at step 10 of Section 4.6.2, "Deploying RTD.ear to Cluster (Detailed option)."

4.7.2 Mapping Roles to Groups After RTD.ear Deployment

If you chose to perform a Fast Path deployment of RTD.ear, then you can check mapping between the standard Oracle RTD roles to WebSphere groups after you have deployed Oracle RTD.

To start this procedure after the deployment of Oracle RTD:

1. Click OracleRTD in Enterprise Applications screen.



Enterprise Applications

Enterprise Applications > **OracleRTD**

Use this page to configure an enterprise application. Click the links to access pages for further configuring of the application or its modules.

Configuration

General Properties

* Name
OracleRTD

Application reference validation
Issue warnings

Detail Properties

- Target specific application status
- Startup behavior
- Application binaries
- Class loading and update detection
- Request dispatcher properties
- Security role to user/group mapping
- JASPI provider
- Custom properties
- View Deployment Descriptor
- Last participant support extension

References

- Resource references
- Shared library references
- Shared library relationships

Modules

- Manage Modules
- Display module build Ids

Web Module Properties

- Session management
- Context Root For Web Modules
- Initialize parameters for servlets
- JSP and JSF options
- Virtual hosts

Enterprise Java Bean Properties

- Default messaging provider references

Client Module Properties

- Client module deployment mode

Database Profiles

- SQLJ profiles and pureQuery bind files

2. Click **Security role to user/group mapping**.

   This displays the same screen **Map security roles to users or groups** that appears at Step 10 (Map security roles to users or groups) of the Detailed deployment of Oracle RTD.

   Depending on what appears in that screen, you must follow one of the processes described in Section 4.7, "Mapping Oracle RTD Roles to WebSphere Groups."

   As described in that section, you must perform either the instructions in Section 4.7.1, "Changing of Roles to Groups Mapping" or Section 4.7.2, "Mapping Roles to Groups After RTD.ear Deployment."

4.8 Creating Custom Roles and Assigning Permissions to Custom Roles (Optional)

This section consists of the following topics:

- Section 4.8.1, "Creating Custom Roles"
- Section 4.8.2, "Assigning Permissions to Custom Roles"

4.8.1 Creating Custom Roles

To create custom roles for Oracle RTD, perform the following high-level steps:

1. Create groups in WebSphere.

To create groups in WebSphere, follow the instructions in Section 4.5.8.1, "Creating Groups" using group names of your own choice.

2. Specify the roles in the deployment descriptor file `application.xml`, extracted from the Oracle file `RTD.ear`. See Section 4.8.1.1, "Specifying Roles in application.xml" for details.
3. Map the roles to the WebSphere groups in the Integrated Solutions Console. See Section 4.8.1.2, "Mapping Roles to Groups" for details.

Then, perform either of the following:

1. Uninstall, then redeploy Oracle RTD, as follows:
   a. Download the deployment descriptor file, `application.xml` back into `RTD.ear`.
   b. Redeploy Oracle RTD using the updated `RTD.ear`. Use Uninstall, then Install.
2. Redeploy Oracle RTD, using Update.

The rest of this section consists of the following topics:

- Section 4.8.1.1, "Specifying Roles in application.xml"
- Section 4.8.1.2, "Mapping Roles to Groups"

4.8.1.1 Specifying Roles in application.xml

To serve as an example, this section describes the addition of a new role, **ILS2Users**.

After extracting `RTD.ear` from *RTD_HOME*`\package`, edit the file `META-INF\application.xml` as follows:

1. Add an entry similar to the following:

```
<security-role>
   <role-name>ILS2Users</role-name>
</security-role>
```

2. Repeat step 1 for as many roles as you want to create.
3. Update RTD.ear with the new application.xml.

4.8.1.2 Mapping Roles to Groups

1. In the Integrated Solutions Console, expand **Applications**, then choose **Enterprise Applications**.
2. Click **OracleRTD**.
3. Under the Detail Properties section, click **Security role to user/group mapping**.
4. Check the role to be modified, then click **Look up groups** and change the mapping.
5. Repeat step 4 for as many roles as you need to map to groups.

4.8.2 Assigning Permissions to Custom Roles

As described in Section 5.4, "Assigning Permissions," assign Cluster permissions, Inline Service permissions, and Decision Center Perspective permissions to any custom roles.

4.9 Creating startJConsole Batch Files

This process consists of getting bootstrap addresses for each server, then creating the JConsole startup script for each server.

This section consists of the following topics:

- Section 4.9.1, "Getting Bootstrap Addresses For Each Server"
- Section 4.9.2, "Setting Up a JConsole Batch File for Windows"
- Section 4.9.3, "Setting Up a JConsole Batch File for Linux Systems"

4.9.1 Getting Bootstrap Addresses For Each Server

To get bootstrap addresses for each server, perform the following steps:

1. In the tree to the left, expand **Servers** and select **Application Servers**.
2. Click **RTDServer1**.
3. Under Communications, expand **Ports**.
4. Write down the BOOTSTRAP_ADDRESS for this server.

   This is used to connect with JConsole for server specific configuration.

4.9.2 Setting Up a JConsole Batch File for Windows

On Windows operating systems, for each server, create a batch file named `startJConsole.bat` and include a file similar to the following *(you must specify your own values for WAS_BOOTSTRAP_PORT and file directories)*:

```
@echo off
@rem WAS_HOME should point to actual directory where WAS AS is installed
set WAS_HOME=C:\IBM\WebSphere\AppServer\
@rem PROFILE_HOME is where your local node profile is; deployment manager profile
is a good place
set PROFILE_HOME=C:\IBM\WebSphere\AppServer\profiles\rtd_dmgr01
set WAS_HOST=localhost
set WAS_BOOTSTRAP_PORT=2812
set CP=%WAS_HOME%/runtimes/com.ibm.ws.admin.client_8.0.0.jar
set CP=%CP%:%WAS_HOME%/runtimes/com.ibm.ws.ejb.thinclient_8.0.0.jar
set CP=%CP%:%WAS_HOME%/runtimes/com.ibm.ws.orb_8.0.0.jar
set CP=%CP%:%WAS_HOME%/java/lib/tools.jar
set CP=%CP%:%WAS_HOME%/java/lib/jconsole.jar
set SSL_PROPS=-J-Dcom.ibm.CORBA.ConfigURL=file:%PROFILE_
HOME%/properties/sas.client.props
set SSL_PROPS=%SSL_PROPS% -J-Dcom.ibm.SSL.ConfigURL=file:%PROFILE_
HOME%/properties/ssl.client.props
%WAS_HOME%\java\bin\jconsole -J-Djava.class.path=%CP% %SSL_PROPS%
service:jmx:iiop://%WAS_HOST%:%WAS_BOOTSTRAP_PORT%/jndi/JMXConnector
```

Tip: Ensure that the JConsole command at the end of this file (`"%WAS_HOME%\java\bin\jconsole..."`) is all on one line.

4.9.3 Setting Up a JConsole Batch File for Linux Systems

On Linux systems, for each server, create a shell script named `startJConsole.sh` similar to the following *(you must specify your own values for WAS_BOOTSTRAP_PORT and file directories)*:

```
#!/bin/sh
# WAS_HOME should point to actual directory where WAS AS is installed
WAS_HOME=/opt/IBM/WebSphere/AppServer/
# PROFILE_HOME is where your local node profile is; deployment manager profile is
a good place
PROFILE_HOME=/opt/IBM/WebSphere/AppServer/profiles/rtd_dmgr01
WAS_HOST=localhost
WAS_BOOTSTRAP_PORT=2809
CP=$WAS_HOME/runtimes/com.ibm.ws.admin.client_8.0.0.jar
CP=$CP:$WAS_HOME/runtimes/com.ibm.ws.ejb.thinclient_8.0.0.jar
CP=$CP:$WAS_HOME/runtimes/com.ibm.ws.orb_8.0.0.jar
CP=$CP:$WAS_HOME/java/lib/tools.jar
CP=$CP:$WAS_HOME/java/lib/jconsole.jar
SAS_PROPS=-J-Dcom.ibm.CORBA.ConfigURL=file:$PROFILE_
HOME/properties/sas.client.props
SSL_PROPS=-J-Dcom.ibm.SSL.ConfigURL=file:$PROFILE_HOME/properties/ssl.client.props
$WAS_HOME/java/bin/jconsole -J-Djava.class.path=$CP $SAS_PROPS $SSL_PROPS
service:jmx:iiop://$WAS_HOST:$WAS_BOOTSTRAP_PORT/jndi/JMXConnector
```

Tip: Ensure that the JConsole command at the end of this file (`$WAS_HOME/java/bin/jconsole...`) is all on one line. Also, ensure the `startJConsole.sh` file has the appropriate execute permissions.

Note: Oracle recommends that you create or edit the file `startJConsole.sh` directly on the Linux system.

If you first create or edit the file on a Windows system, and subsequently transfer it to a Linux system using ftp, then make sure that you use binary transfer node, not ascii.

4.10 Extending the Cluster

Extending the cluster to a second machine (or further) is similar to the initial installation and configuration. In summary, the steps required are the following:

- Install WebSphere Application Server Network Deployment on the second machine
- Create a profile with just a node, no servers
- Make the Deployment manager aware of new node
- Add the cluster member via WebSphere Administrative Console

This section contains the following topics:

- Section 4.10.1, "Preparing the Second Machine"
- Section 4.10.2, "Installing Installation Manager"
- Section 4.10.3, "Running Installation Manager"
- Section 4.10.4, "Running the Profile Management Tool"
- Section 4.10.5, "Adding New Node to Existing Cell"
- Section 4.10.6, "Creating Additional Server in the Cluster"
- Section 4.10.7, "Fixing Properties on the Additional Server"

4.10.1 Preparing the Second Machine

Copy all the WebSphere source files to the second machine, and unpack them into separate directories as on the first machine.

4.10.2 Installing Installation Manager

As on the first machine, install Installation Manager.

4.10.3 Running Installation Manager

When running Installation Manager, in the Install Packages screen, select only IBM WebSphere Application Server Network Deployment (and any interim fixes recommended by IBM).

4.10.4 Running the Profile Management Tool

1. Start the Profile Management Tool by running `{WAS_HOME}/WebSphere/AppServer/bin/ProfileManagement/pmt.sh` (or the equivalent .bat file on a Windows machine).
2. Create a new Custom profile (not a Cell profile, as on the first machine).
3. Select "Advanced profile creation", name the profile *(for example, rtd_node2)* and select its location *(for example, /scratch/IBM/WebSphere/AppServer/profiles/rtd_node2*).
4. For the Node and Host Names, name the new node *(for example, rtd_cell_node02)*, and ensure that the Host Name is either the DNS name or the IP address of the second machine.
5. On the Federation screen, select **Federate this node later**.
6. Accept the defaults for SSL certificate generation and port assignments, then create the profile.
7. Exit the Profile Management Tool.

4.10.5 Adding New Node to Existing Cell

To add the new node to the existing cell on the first machine:

1. Check what SOAP port deployment manager is on:

   System administration > Deployment manager, then, under **Additional Properties > Ports**, look up the port for SOAP_CONNECTOR_ADDRESS.

2. Using this port and host name of machine 1, run the following command (or its Windows .bat equivalent):

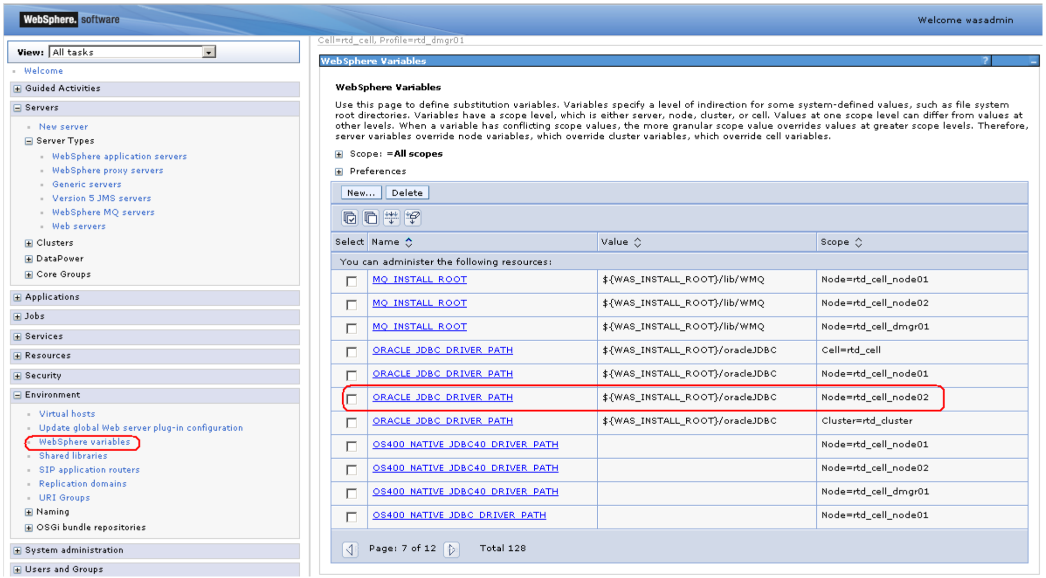
   - *WAS_HOME*/bin/addNode.sh {host} {port} -username {user} -password {password}

   where {host} and {port} define the machine where the Deployment manager is listening, and {user} and {password} are administrative credentials, and *WAS_HOME* typically has a sub-directory path like .../IBM/WebSphere/AppServer.

   When this completes successfully and the node is federated, the cell is extended and includes the additional node, as displayed in the **System administration > Nodes** screen. Note that the Host Name for the new node displays the name (or IP Address) of the second machine.

3. Copy ojdbc6.jar driver to the new machine and fix the JDBC WebSphere path variable ORACLE_JDBC_DRIVER_PATH.

   Typically the directory path for ojdbc6.jar will be identical on both the source and target machines. When you add a node, ORACLE_JDBC_DRIVER_PATH is blank for the new node. So, after copying ojdbc6.jar to the new machine, edit ORACLE_JDBC_DRIVER_PATH there, specifying the correct path.



WebSphere Variables

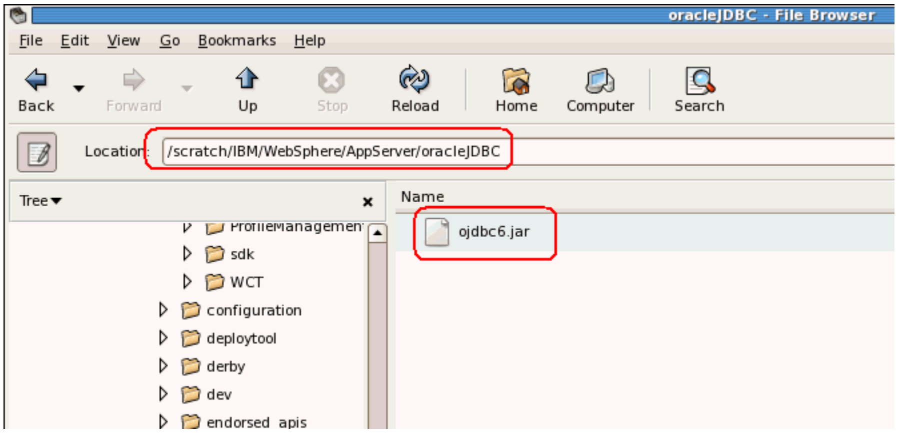
Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root directories. Variables have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable has conflicting scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which override cluster variables, which override cell variables.

Scope: =**All scopes**

Preferences

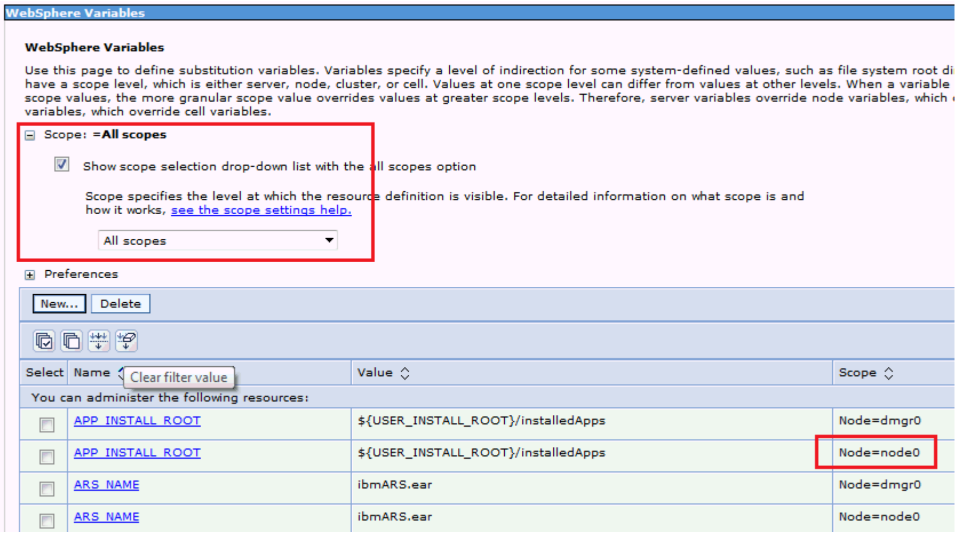
Select	Name	Value	Scope
You can administer the following resources:			
☐	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=rtd_cell_node01
☐	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=rtd_cell_node02
☐	MQ_INSTALL_ROOT	${WAS_INSTALL_ROOT}/lib/WMQ	Node=rtd_cell_dmgr01
☐	ORACLE_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/oracleJDBC	Cell=rtd_cell
☐	ORACLE_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/oracleJDBC	Node=rtd_cell_node01
☐	ORACLE_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/oracleJDBC	Node=rtd_cell_node02
☐	ORACLE_JDBC_DRIVER_PATH	${WAS_INSTALL_ROOT}/oracleJDBC	Cluster=rtd_cluster
☐	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=rtd_cell_node01
☐	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=rtd_cell_node02
☐	OS400_NATIVE_JDBC40_DRIVER_PATH		Node=rtd_cell_dmgr01
☐	OS400_NATIVE_JDBC_DRIVER_PATH		Node=rtd_cell_node01

Page: 7 of 12 Total 128



Location: /scratch/IBM/WebSphere/AppServer/oracleJDBC

When working with WebSphere variables, make sure that you have selected the correct scope or select "All scopes" and look at the scope column. There could be several ORACLE_JDBC_DRIVER_PATH variables in different scopes. Change the variable in the appropriate scope only.



WebSphere Variables

Use this page to define substitution variables. Variables specify a level of indirection for some system-defined values, such as file system root di have a scope level, which is either server, node, cluster, or cell. Values at one scope level can differ from values at other levels. When a variable scope values, the more granular scope value overrides values at greater scope levels. Therefore, server variables override node variables, which variables, which override cell variables.

Scope: =**All scopes**

☑ Show scope selection drop-down list with the all scopes option

Scope specifies the level at which the resource definition is visible. For detailed information on what scope is and how it works, see the scope settings help.

All scopes

Preferences

Select	Name	Value	Scope
You can administer the following resources:			
☐	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps	Node=dmgr0
☐	APP_INSTALL_ROOT	${USER_INSTALL_ROOT}/installedApps	Node=node0
☐	ARS_NAME	ibmARS.ear	Node=dmgr0
☐	ARS_NAME	ibmARS.ear	Node=node0

4.10.6 Creating Additional Server in the Cluster

To extend the cluster, and ensure that the new server is located o the second node, perform the following steps:

1. Select Clusters > WebSphere application server clusters > rtd_cluster > (Cluster members) > New.
2. For Member name, enter **rtd_server2**, for Select node, select the new node *(for example, rtd_cell_node02)*, then click Add Member.

   The new member appears in the list at the foot of the screen.
3. Click **Next > Finish.**

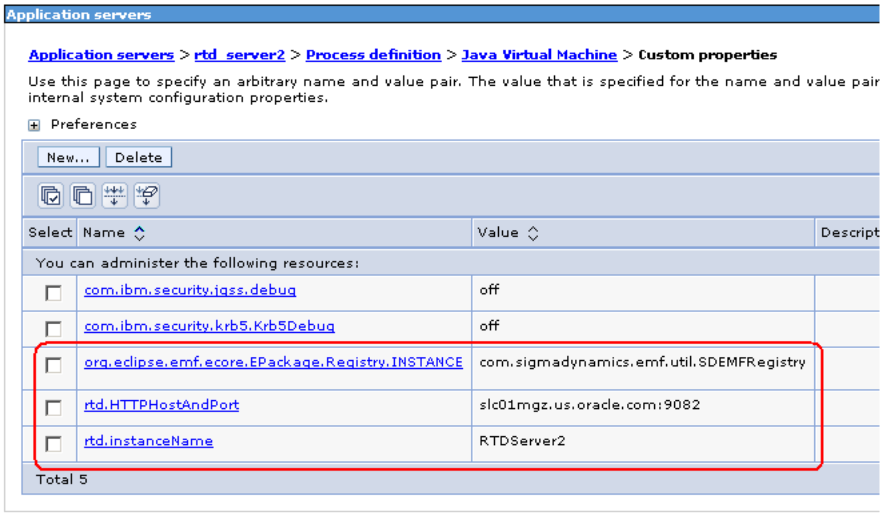
The cluster has been extended - see **WebSphere application server clusters > rtd_cluster > (Cluster members)**.

Because OracleRTD is targeted to the cluster, OracleRTD is deployed on the second server automatically - it appears in **WebSphere application server clusters > rtd_cluster > Cluster members > rtd_server2 > Installed applications**.

4.10.7 Fixing Properties on the Additional Server

To fix the properties on the additional server, perform the following steps:

1. Enter or change the properties for the second server in the following screens:



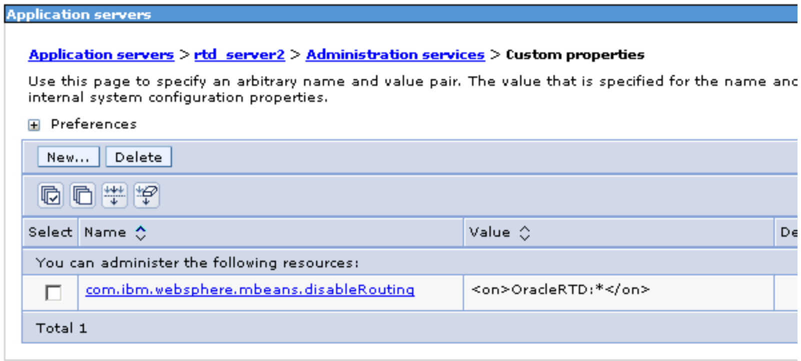
Application servers

Application servers > rtd_server2 > Process definition > Java Virtual Machine > Custom properties

Use this page to specify an arbitrary name and value pair. The value that is specified for the name and value pair
internal system configuration properties.

Select	Name	Value	Descript
You can administer the following resources:			
☐	com.ibm.security.jgss.debug	off	
☐	com.ibm.security.krb5.Krb5Debug	off	
☐	org.eclipse.emf.ecore.EPackage.Registry.INSTANCE	com.sigmadynamics.emf.util.SDEMFRegistry	
☐	rtd.HTTPHostAndPort	slc01mgz.us.oracle.com:9082	
☐	rtd.instanceName	RTDServer2	

Total 5



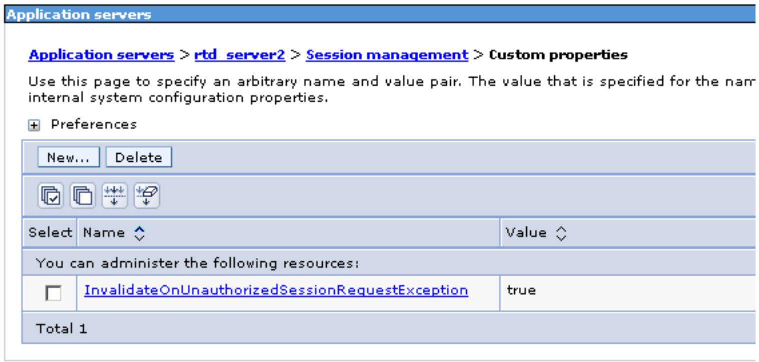
Application servers

Application servers > rtd_server2 > Administration services > Custom properties

Use this page to specify an arbitrary name and value pair. The value that is specified for the name anc
internal system configuration properties.

Select	Name	Value	De
You can administer the following resources:			
☐	com.ibm.websphere.mbeans.disableRouting	<on>OracleRTD:*</on>	

Total 1



Application servers

Application servers > rtd_server2 > Session management > Custom properties

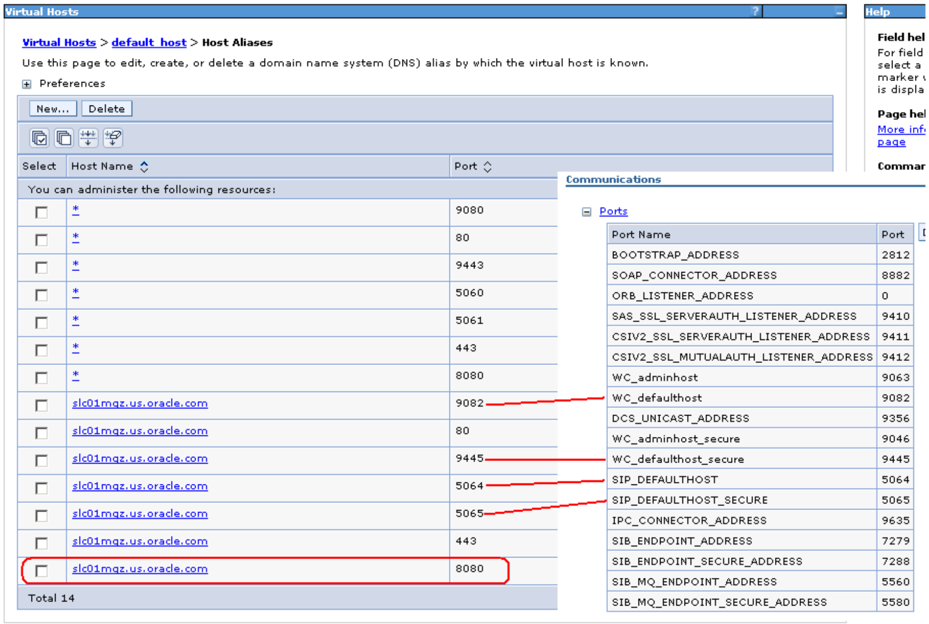
Use this page to specify an arbitrary name and value pair. The value that is specified for the narr
internal system configuration properties.

Select	Name	Value
You can administer the following resources:		
☐	InvalidateOnUnauthorizedSessionRequestException	true

Total 1

Note: only rtd.HTTPHostAndPort and rtd.instanceName have to change, the rest are the same.

2. Add the default host alias for the second node and make sure that the ports on the second machine match the default host aliases.



Virtual Hosts > default_host > Host Aliases

Use this page to edit, create, or delete a domain name system (DNS) alias by which the virtual host is known.

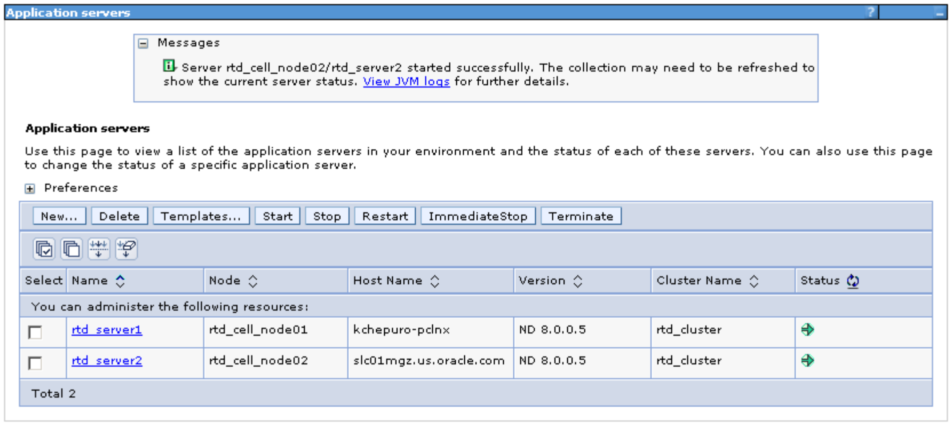
⊞ Preferences

Select	Host Name	Port
You can administer the following resources:		
☐	*	9080
☐	*	80
☐	*	9443
☐	*	5060
☐	*	5061
☐	*	443
☐	*	8080
☐	slc01mqz.us.oracle.com	9082
☐	slc01mqz.us.oracle.com	80
☐	slc01mqz.us.oracle.com	9445
☐	slc01mqz.us.oracle.com	5064
☐	slc01mqz.us.oracle.com	5065
☐	slc01mqz.us.oracle.com	443
☐	slc01mqz.us.oracle.com	8080
Total 14		

Port Name	Port
BOOTSTRAP_ADDRESS	2812
SOAP_CONNECTOR_ADDRESS	8882
ORB_LISTENER_ADDRESS	0
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9410
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9411
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9412
WC_adminhost	9063
WC_defaulthost	9082
DCS_UNICAST_ADDRESS	9356
WC_adminhost_secure	9046
WC_defaulthost_secure	9445
SIP_DEFAULTHOST	5064
SIP_DEFAULTHOST_SECURE	5065
IPC_CONNECTOR_ADDRESS	9635
SIB_ENDPOINT_ADDRESS	7279
SIB_ENDPOINT_SECURE_ADDRESS	7288
SIB_MQ_ENDPOINT_ADDRESS	5560
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5580

If they are not, fix the Host aliases to use the actual ports that the server is using.

3. Now start rtd_server2.



⊟ Messages

Server rtd_cell_node02/rtd_server2 started successfully. The collection may need to be refreshed to show the current server status. View JVM logs for further details.

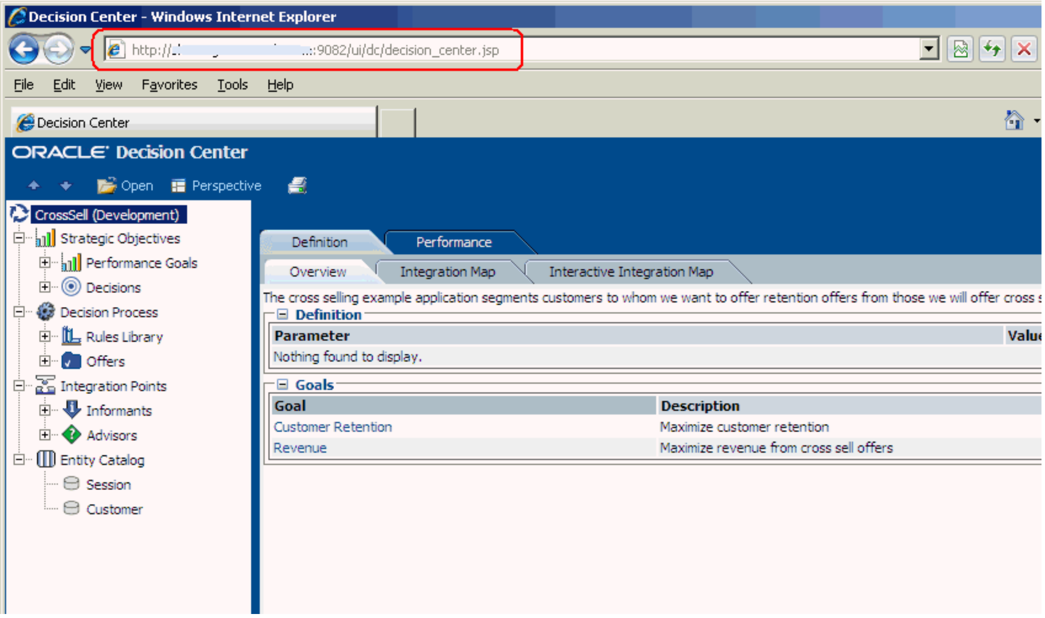
Application servers

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.

⊞ Preferences

Select	Name	Node	Host Name	Version	Cluster Name	Status
You can administer the following resources:						
☐	rtd_server1	rtd_cell_node01	kchepuro-pclnx	ND 8.0.0.5	rtd_cluster	➔
☐	rtd_server2	rtd_cell_node02	slc01mgz.us.oracle.com	ND 8.0.0.5	rtd_cluster	➔
Total 2						

4. Verify that Oracle RTD is running on the second machine, for example by running Decision Center.



The cross selling example application segments customers to whom we want to offer retention offers from those we will offer cross s

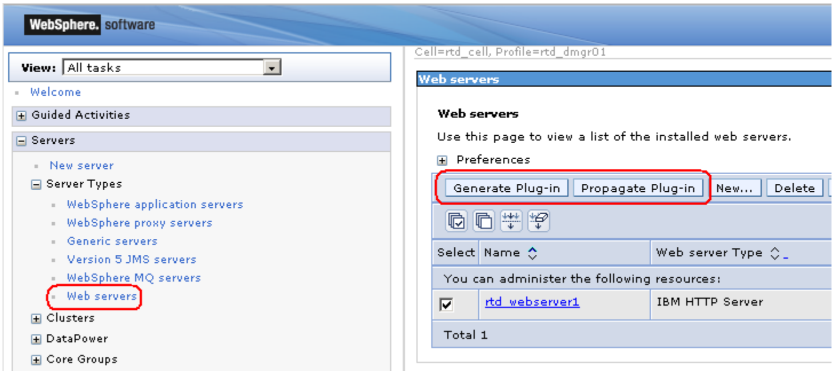
Definition

Parameter	Valu
Nothing found to display.	

Goals

Goal	Description
Customer Retention	Maximize customer retention
Revenue	Maximize revenue from cross sell offers

5. Ensure that you generate and propagate Web Server plug-in:



Web servers

Use this page to view a list of the installed web servers.

Select	Name	Web server Type
You can administer the following resources:		
☑	rtd_webserver1	IBM HTTP Server
Total 1		

6. Ensure that Oracle RTD is accessible via HTTP server port 8080 with requests routed to both cluster servers

4.11 Configuring New Data Sources

Configuring a new data source consists of the following steps:

- Creating J2C Authentication Alias - this specifies how to login to your database
- Creating JDBC Data Source - this enables Oracle RTD to find and access the database

---

Note: It is possible that you may need an additional step to create a new JDBC provider, depending on user requirements, but this section assumes that you are reusing the existing JDBC provider you created in Section 4.5.7.2, "Creating JDBC Provider."

Testing the new J2C authentication alias and data source consists of the following steps:

- Restarting the Node
- Testing New Data Source Connection

4.11.1 Creating J2C Authentication Alias

To create a J2C authentication alias for the new data source, perform the following steps:

> **Note:** Unless you must have compatibility with earlier releases, you do not need to select the option **Prefix new alias names with the node name of the cell**.

Table 4–21 Creating J2C Authentication Alias

Access path or Screen area	Targeted objects	Summary Description
Security > **Global security >** **Java Authentication and Authorization Service** > **J2C authentication data >** **New**	J2C authentication alias	Enter the Alias name. Enter UserId and Password, for the database runtime user.

4.11.2 Creating JDBC Data Source

To create the new data source, perform the following steps:

Table 4–22 Creating Data Source

Access path or Screen area	Targeted objects	Summary Description
Resources > **JDBC** > **Data sources >** **New**	Data source for SDDS	**1.** Scope: Select cluster scope **Data source name** can be any name **JNDI name** can be any name (except SDDS) Click **Next**. **2.** **Select an existing JDBC provider**: Select the JDBC provider that you created in Section 4.5.7.2, "Creating JDBC Provider." Click **Next**. **3.** **URL:** Enter the URL using the format: *jdbc:oracle:thin:@{database_host}:{database_port}:{database_name}* For example, jdbc:oracle:thin:@myhost:1521:orcl **Data store helper class name:** Oracle11g data store helper Click **Next**. **4.** **Component-managed authentication alias**: Select the alias that you created in Section 4.11.1, "Creating J2C Authentication Alias." Keep the default settings of None for **Mapping-configuration alias** and **Container-manager authentication alias**. **Next > Finish**.

4.11.3 Restarting the Node

To test your new data source, you must restart the WebSphere managed node, as described in Section 4.5.7.5, "Restarting the Node." This node restart automatically activates the new J2C authentication alias.

4.11.4 Testing New Data Source Connection

To test the new data source connection, perform the following steps:

Table 4–23 Testing Data Source Connection

Access path or Screen area	Targeted objects	Summary Description
Resources > **JDBC** > **Data sources**	New data source	Select the data source, then, for Test Table, enter the name of an existing table in the database. Click **Test connection**.

4.12 Configuring SSL for Real-Time Decision Server (Recommended)

Follow the steps in this section to set up SSL for all client connections to Real-Time Decision Server. Before you begin, ensure that you followed the instructions in Section 2.5, "Using SSL with Oracle Real-Time Decisions" to change the default Oracle RTD keystore and truststore passwords. Also, ensure that WebSphere is started.

> **Note:** If you want to use your own keystore and truststore, you do not need to complete the instructions in Section 2.5.

To configure SSL for Real-Time Decision Server:

1. Access the Integrated Solutions Console at the URL `http://`*`websphere_host`*`:`*`port`*`/ibm/console`. At the login prompt, enter the administrator user name and password. On Windows, you can also access the Integrated Solutions Console through **Start > Programs**.
2. In the tree on the left, expand **Security** and choose **SSL certificate and key management**.
3. Under the Related Items heading, click **Key stores and certificates**.
4. Create the Oracle RTD keystore, as follows:
   a. Click **New**.
   b. For **Name**, enter `OracleRTD_KeyStore`.
   c. For **Path**, enter *`RTD_HOME`*`/etc/ssl/sdserver.keystore`. Alternatively, if you do not want to use the default Oracle RTD keystore, enter the path to your own keystore.
   d. For **Password** and **Confirm password**, enter the password for your keystore. If you are using the default Oracle RTD keystore, enter the password you created in Section 2.6.
   e. For **Type**, select `JKS`.
   f. Click **OK**.
5. Create the Oracle RTD truststore, as follows:
   a. On the Key stores and certificates page, click **New**.

   b. For **Name**, enter `OracleRTD_TrustStore`.
   c. For **Path**, enter *`RTD_HOME`*`/etc/ssl/sdtrust.store`. Alternatively, if you do not want to use the default Oracle RTD truststore, enter the path to your own truststore.
   d. For **Password** and **Confirm password**, enter the password for your truststore. If you are using the default Oracle RTD truststore, enter the password you created in Section 2.6.
   e. For **Type**, select **JKS**.
   f. Click **OK**.

6. Return to the SSL certificate and key management page and create an SSL configuration for Oracle RTD, as follows:
   a. Under the Related Items heading, click **SSL configurations**.
   b. Click **New**.
   c. For **Name**, enter `OracleRTD_SSL`.
   d. For **Trust store name**, select **OracleRTD_TrustStore**.
   e. For **Keystore name**, select **OracleRTD_KeyStore**.
   f. Click **Get certificate aliases**.
   g. Click **OK**.
7. Set the Transport Chain HTTPS port, as follows:
   a. In the tree on the left, expand **Servers** and choose **Application servers**.
   b. Click the name of the application server where you have installed Oracle RTD (for example, **server1**).
   c. Under the Container Settings heading, expand **Web Container Settings** and choose **Web container transport chains**.
   d. Click **WCInboundDefaultSecure**.
   e. Under the Transport Channels heading, click the name of the TCP inbound channel (for example, **TCP inbound channel (TCP 4)**).
   f. Under the Related Items heading, click **Ports**.
   g. Click **WC_defaulthost_secure**.
   h. Change the **Port** to `8443`.
   i. Click **OK**, then click **Save**.
8. Set the Virtual Host HTTPS Port, as follows:
   a. In the tree on the left, expand **Environment** and choose **Virtual Hosts**.
   b. Click **default_host**.
   c. Under the Additional Properties heading, click **Host Aliases**.
   d. For port **9443**, click *****.
   e. Change the **Port** to `8443`.
   f. Click **OK**.
9. Set the HTTPS port to use the Oracle RTD SSL configuration you created in Step 6, as follows:

   a. In the tree on the left, expand **Security** and choose **SSL certificate and key management**.
   b. Under the Configuration settings heading, click **Manage endpoint security configurations**.
   c. Under the Local Topology heading, expand **Inbound** > *cell_name* > **nodes** > *node_name* > **servers** > *server_name*, then click **WC_defaulthost_secure**.
   d. Under the heading Specific SSL configuration for this endpoint, select **Override inherited values**, then select **OracleRTD_SSL** for **SSL configuration**.
   e. Click **Update certificate alias list**.
   f. For **Certificate alias in key store**, select your keystore password.
   g. Click **Apply**.

10. Click **Save**.
11. Restart WebSphere.

> **Note:** For a truly secure environment, you should also disable the regular HTTP port to ensure that all client connections are routed through the SSL port. To do this, perform the following step:
>
> - Disable the HTTP port for your Web server using application server tools. Refer to the WebSphere documentation for more information.

12. If you are using your own keystore and truststore, perform the following additional steps to enable SSL for Decision Studio. You do not need to perform these steps if you are using the default Oracle RTD keystore and truststore.
    a. Open `RTD_HOME\eclipse\eclipse.ini` for editing.
    b. Locate the following line:

    ```
    -Djava.net.ssl.trustStore="..\etc\ssl\sdtrust.store"
    ```

    c. Replace `..\etc\ssl\sdtruststore` with the full path to your truststore file.
    d. Save and close the file.
    e. Open `RTD_HOME\scripts\sdexec.cmd` for editing.
    f. Locate the line beginning with `%SD_START%`, near the bottom of the file. Near the end of the line, locate the following string:

    ```
    -Djavax.net.ssl.trustStore="%SD_ROOT%\etc\ssl\sdtrust.store"
    ```

    g. Replace `%SD_ROOT%\etc\ssl\sdtruststore` with the full path to your truststore file.
    h. Save and close the file.

4.12.1 Testing the SSL Configuration

To verify that the SSL port is functioning properly, go to Decision Center at the URL `https://server_name:ssl_port/ui`. If the SSL port is functioning property, your browser will display the "Welcome to Decision Center" login screen.

You may get a message from your Web browser, similar to "Do you want to accept this certificate?" This message is generated because the browser does not know about the self-signed certificate that was shipped with the default Oracle RTD keystore. This self-signed certificate is suitable for development and test environments, but it is not recommended for production environments.

For production environments, Oracle recommends the self-signed certificate be replaced with a certificate from a trusted certificate authority (CA), like Verisign/Thawte, by submitting to the CA a certificate request generated by Sun's keytool utility.

For instructions on generating a certificate request, and for importing the certificate from the CA into the keystore, go to one of the following URLs:

For Linux and Solaris:

`http://docs.oracle.com/javase/7/docs/technotes/tools/solaris/keytool.html`

For Windows:

`http://docs.oracle.com/javase/7/docs/technotes/tools/windows/keytool.html`

5

Configuring Security for Oracle Real-Time Decisions

This chapter provides an overview of the security framework that enables and restricts access to Oracle RTD and the different components of Oracle RTD.

Oracle RTD has three different levels of security: cluster-wide security, Inline Service (application-level) security, and an additional level of security for perspectives (views) in Decision Center. Also, each data source has an additional level of security, based on the database user name and password.

The setup of each Oracle RTD security level can be summarized as follows:

- Cluster-wide security is configured using the SecurityManager MBean. You can use JConsole to set permission codes for Oracle RTD users.
- Inline Service security is configured on individual Inline Services.
- Perspective-level security is configured in Decision Studio.

The sections that follow describe the setup and usage of each of the security levels in detail.

It is important to configure all three levels of security. Otherwise, you may run into problems. For example, if a user has cluster-wide permission to deploy in Decision Studio, but does not have the correct permissions to do so at the Inline Service and perspective level, the user will not be able to view and edit the Inline Service.

This chapter contains the following topics:

- Section 5.1, "Permissions"
- Section 5.2, "Standard Oracle RTD Roles"
- Section 5.3, "Custom Roles"
- Section 5.4, "Assigning Permissions"
- Section 5.5, "Debugging Role Assignments"
- Section 5.6, "Migration Strategy"

5.1 Permissions

There are three types of permission used in Oracle RTD:

- Cluster permissions
- Inline Service permissions
- Decision Center Perspective permissions

This section consists of the following topics:

- Section 5.1.1, "Cluster Permissions"
- Section 5.1.2, "Inline Service Permissions"
- Section 5.1.3, "Decision Center Perspective Permissions"

5.1.1 Cluster Permissions

Cluster permissions are global permissions that apply to all Oracle RTD instances of the Oracle RTD cluster.

Oracle RTD cluster permissions are granted to an individual JEE Servlet Role through the JMX MBean whose JConsole path is OracleRTD > SDManagement > Security Manager.

Table 5–1 Cluster Permissions

Target Cluster Permission	Description	Implied Permissions, Automatically Granted when Granting the Target Permission
Administrator	Authorizes all Oracle RTD activities, including the ability to work with Inline Services to which no Inline Service permissions have been assigned.	Open Service Open Service For Reading Deploy Service From Studio Download Service
Open Service	Authorizes a person to use Decision Center to open an Inline Service for editing, if the Inline Service permission Open Service has also been granted to the person.	Open Service For Reading
Open Service For Reading	Authorizes a person to use Decision Center to open an Inline Service for viewing, if the Inline Service permission Open Service For Reading has also been granted to the person.	N/A
Deploy Service From Studio	Authorizes a person to deploy a new Inline Service from Decision Studio, or to deploy a new version of an existing Inline Service, if the new Inline Service and the existing version (if any) have both been granted the Inline Service permission Deploy Service From Studio.	Open Service Open Service For Reading
Download Service	Authorizes a person to use Decision Studio to download an Inline Service from a server if the Inline Service has been granted the Inline Service permission Download Service.	Open Service For Reading

5.1.2 Inline Service Permissions

Inline Service permissions are associated with a specific Inline Service. These are granted to an individual Servlet Role through Decision Studio's Application/Permissions tab.

Table 5–2 Inline Service Permissions

Target Permission	Description	Implied Permissions, Automatically Granted when Granting the Target Permission
Open Service	When accompanied by the corresponding cluster permission, authorizes a person to use Decision Center to open an Inline Service for editing.	Open Service For Reading
Open Service For Reading	When accompanied by the corresponding cluster permission, authorizes a person to use Decision Center to open an Inline Service for viewing.	N/A
Deploy Service From Studio	When accompanied by the corresponding cluster permission, authorizes a person to deploy the Inline Service from Decision Studio. If redeploying an existing Inline Service, the existing Inline Service must also have this permission.	Open Service Open Service For Reading
Download Service	When accompanied by the corresponding cluster permission, authorizes a person to use Decision Studio to download the Inline Service from a server.	Open Service For Reading

5.1.3 Decision Center Perspective Permissions

Decision Center Perspective permissions are assigned through Decision Studio, to enable a Decision Center user to open the Inline Service using the specified perspective.

Table 5–3 lists the default Decision Center perspectives.

Table 5–3 Decision Center Perspectives

Perspective	Description
Explore	Presents a user interface suitable for viewing attributes and performance reports for an Inline Service, but not for editing.
Design	Presents a user interface that includes all of the Explore perspective, and also the ability to edit some aspects of the Inline Service.
At a Glance	Presents an abbreviated user interface suitable for viewing a subset of the attributes and performance reports that are available through the Explore perspective.

5.2 Standard Oracle RTD Roles

This section describes the JEE Servlet Roles declared in Oracle RTD's deployment descriptors, out of the box, and the Oracle RTD permissions that Oracle RTD automatically assigns to them.

These roles are not created by Oracle RTD, but merely referenced by Oracle RTD's deployment descriptors. To actually create them, the application server's administration console would typically be used to first create user groups, and then to define roles in terms of the user groups, as described in separate sections in this chapter.

This section consists of the following topics:

- Section 5.2.1, "Default Cluster Permission Assignments"
- Section 5.2.2, "Default Inline Service Permissions"
- Section 5.2.3, "Default Decision Center Perspective Permissions"

5.2.1 Default Cluster Permission Assignments

This section describes the default assignments of cluster permissions to standard Oracle RTD roles.

You can use Oracle RTD's Security Manager MBean to change these assignments, but this is not recommended.

> **Note:** There is an MBean operation that reverts to this default assignment of cluster permissions, namely, OracleRTD > SDManagement > Security Manager: revertToStandardPermissions().

Table 5–4 Standard Oracle RTD Roles

Role	**Default Cluster Permissions**	**Description** ***(with references to example Inline Service ILS_X)***
RTDUsers	None	This role corresponds to all authenticated users. Requests sent to any Decision Center pages or web services must be in this role to be granted access by the JEE container. Because no permissions are assigned to this role, Oracle RTD would subsequently reject the request when it checks for required permissions, unless the user is also in some other role that has the required permissions.
RTDAdministrators	Administrator	This role is allowed to do anything.
RTDDecisionCenterEditors	Open Service	This role may use Decision Center to open Inline Service *ILS_X* for editing, if *ILS_X*'s Application/Permissions tab in Decision Studio grants the Open Service permission to this role.
RTDDecisionCenterUsers	Open Service For Reading	This role may use Decision Center to open Inline Service *ILS_X* for read-only viewing, if *ILS_X*'s Application/Permissions tab in Decision Studio grants the Open Service For Reading permission to this role.
RTDStudioDeployers	Deploy Service From Studio Open Service	This role may use Decision Studio to deploy Inline Service *ILS_X* to a server if *ILS_X*'s Application/Permissions tab in Decision Studio grants the Deploy Service From Studio permission, and an existing instance of *ILS_X* has not been deployed that does not have this permission. This role may use Decision Center to open Inline Service *ILS_X* for editing, if *ILS_X*'s Application/Permissions tab in Decision Studio grants the Open Service permission to this role.
RTDStudioDownloaders	Download Service Open Service For Reading	This role may use Decision Studio to download Inline Service *ILS_X* from a server if *ILS_X*'s Application/Permissions tab in Decision Studio granted the Download Service permission to this role before *ILS_X* was deployed. This role may use Decision Center to open Inline Service *ILS_X* for read-only viewing if *ILS_X*'s Application/Permissions tab in Decision Studio grants the Open Service For Reading permission to this role.
RTDBatchAdministrators	None	This role may execute any methods of the BatchManager web service. No cluster permissions are associated with this role.
RTDChoiceEditors	None	This role may execute any methods of the ExternalChoice web service, and may access Decision Center's URLs serving requests to edit an external rule or external choice. No cluster permissions are associated with this role.

5.2.2 Default Inline Service Permissions

By default, a new Inline Service has a number of Inline Service permissions assigned to the standard Oracle RTD roles.

Table 5–5 shows the default Inline Service permissions assigned to the standard RTD roles.

Table 5–5 Default Inline Service Permissions Assigned to Standard Oracle RTD Roles

Standard Role	Inline Service Permissions
RTDDecisionCenterEditors	Open Service Open Service for Reading
RTDDecisionCenterEditors	Open Service Open Service for Reading
RTDDecisionCenterUsers	Open Service
RTDStudioDeployers	Deploy Service from Studio Open Service Open Service for Reading
RTDStudioDownloaders	Download Service Open Service for Reading

5.2.3 Default Decision Center Perspective Permissions

When a new Inline Service is created, its Decision Center perspectives will by default be accessible to the standard Oracle RTD roles, as shown in Table 5–6.

Table 5–6 Decision Center Perspective Permissions

Perspective	Granted to Role
Explore	RTDDecisionCenterUsers RTDDecisionCenterEditors RTDStudioDeployers RTDStudioDownloaders
Design	RTDDecisionCenterEditors RTDStudioDeployers
At a Glance	RTDDecisionCenterUsers RTDDecisionCenterEditors RTDStudioDeployers RTDStudioDownloaders

5.3 Custom Roles

If an enterprise needs to assign permissions to more or different groups of users than defined by Oracle RTD's standard roles, the enterprise can create new security roles corresponding to the specialized user groups, declare the new roles in Oracle RTD deployment descriptors, and redeploy Oracle RTD before granting the Cluster or Inline Service permissions to the new role as described in previous sections.

The details of creating the roles and of declaring them in Oracle RTD's deployment descriptors are vendor-specific, and are described in the application server specific chapters.

As an overview of the process, assume that the enterprise has already defined its first Inline Service, ILS1, and that ILS1 is managed by users in standard Oracle RTD roles. Now, assume that the enterprise has the following extra requirements:

- To introduce a new Inline Service, ILS2, to be managed by users in the groups **ILS2DevelopersGroup** and **ILS2UsersGroup**
- Developers for ILS2 will be allowed to read the contents of ILS1, but not to modify it
- Developers of ILS1 will have no access to ILS2, not even for reading

The following is an overview of the steps for the extra requirements:

1. Use the vendor's administration console to create user groups, **ILS2DevelopersGroup** and **ILS2UsersGroup**.
2. Use the vendor's administration console to put the appropriate users into the two groups.
3. Use the vendor's administration console to create two JEE Servlet security roles, **ILS2Developers** and **ILS2Users**, so that, in terms of the new groups:
   a. Any user in **ILS2DevelopersGroup** is in the **ILS2Developers** role
   b. Any user in **ILS2UsersGroup** is in the **ILS2Users** role
4. Declare references to the new security roles by updating Oracle RTD's deployment descriptors in two `web.xml` files:
   - `RTD.ear` > `ui.war` > `web-inf/web.xml`
   - `RTD.ear` > `soap.war` > `web-inf/web.xml`

   The notation here for the `web.xml` paths signifies that both `web.xml` files are in directories named `web-inf`, and that `web-inf` is inside web archives named `ui.war` and `soap.war`. Similarly, the web archives are inside an enterprise archive named `RTD.ear`.

   Depending on the installation, which varies for different application servers, the `ear` and `war` files may be archive files, or they may have been expanded into the file system.
5. Redeploy Oracle RTD.
6. Use JConsole to assign cluster permissions to the new roles, as follows:
   - Deploy Service from Studio to the role **ILS2Developers**.

     This permission is necessary to allow Inline Service deployment, but insufficient without corresponding Inline Service permissions.
   - Open Service to the role **ILS2Users**.

     This permission is necessary to allow users to open the Inline Service in Decision Center, but insufficient without corresponding Inline Service permissions
7. Use Decision Studio to open ILS1 and ILS2, to assign these Inline Service permissions and Decision Center Perspective permissions, as shown in Table 5–7.

Table 5–7 Inline Service and Decision Center Perspective Permissions

Inline Service	Role	Permissions	Perspectives
ILS1	ILS2Developers	Open Service for Reading	Explore, At a Glance
ILS2	ILS2Developers	Deploy Service from Studio Open Service Open Service for Reading	Explore, Design, At a Glance
ILS2	ILS2Users	Open Service Open Service for Reading	Explore, Design, At a Glance

8. Use Decision Studio to redeploy the two Inline Services, ILS1 and ILS2.

5.4 Assigning Permissions

Section 5.2, "Standard Oracle RTD Roles" and its component subsections describe how default permissions are already assigned to the standard Oracle RTD roles. This section describes how to assign other permissions, such as are required for custom roles, where there are no default permissions.

This section consists of the following topics:

- Section 5.4.1, "Granting Cluster Permissions"
- Section 5.4.2, "Granting Inline Service Permissions"
- Section 5.4.3, "Granting Decision Center Perspective Permissions"

5.4.1 Granting Cluster Permissions

Cluster permissions are granted through JMX, using the MBean whose path is OracleRTD > Cluster > Security Manager. For more details, see Section 11.4.13, "About OracleRTD > Cluster > Security Manager."

5.4.1.1 Operation listPermissionCodes

The **listPermissionCodes** operation provides a list of permission codes that can be entered when assigning permissions.

For example:

- 0 - Administrator
- 1 - Open Service
- 2 - Open Service for Reading
- 3 - Deploy Service from Studio
- 4 - Download Service

5.4.1.2 Operation assignPermission

The **assignPermission** operation accepts two arguments, the name of a role, and the code for the permission to be granted to that role.

For example, Role: ILS2Developers and Code: 3.

5.4.1.3 Operation listDirectPermissions

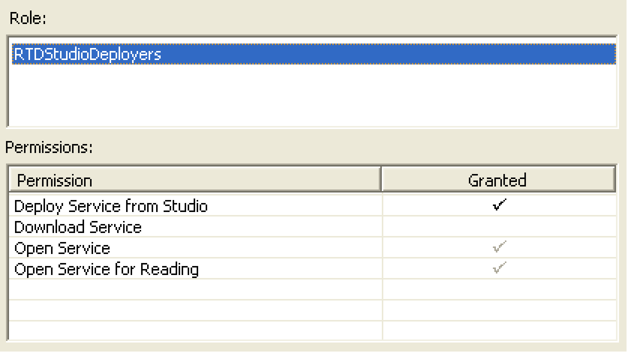
The **listDirectPermissions** operation shows all the cluster permissions that have been granted to a role, including permissions granted directly by the **assignPermission** operation, and any implied permissions.

For example, the permissions granted to the role, ILS2Developers, as a consequence of granting permission code 3 through the **assignPermission** operation, may appear as:

- Deploy Service from Studio
- Open Service
- Open Service for Reading

5.4.2 Granting Inline Service Permissions

Inline Service permissions are granted through Decision Studio, in the Permissions tab of the Application object. For more details, see the section "Setting Inline Service Permissions" in *Oracle Real-Time Decisions Platform Developer's Guide*.



Role:

RTDStudioDeployers

Permissions:

Permission	Granted
Deploy Service from Studio	✓
Download Service	
Open Service	✓
Open Service for Reading	✓

For example, in the CrossSell Inline Service, the RTDStudioDeployers role has been granted the Deploy Service from Studio permission. The Open Service and Open Service for Reading permissions are also marked as granted because these are implied by Deploy Service from Studio.

In Decision Studio, when granting a permission, the name of the role can be selected from a dropdown list of roles to which cluster permissions have already been assigned. If the intended role has not been yet assigned any cluster permissions, you can manually enter the name in Decision Studio, but the permission will have no value until the corresponding cluster permission has been granted.

5.4.3 Granting Decision Center Perspective Permissions

Note: If some role does not have access rights to at least one of the perspectives, the user will not be able to open ILS in browser . Therefore, you should grant at least one perspective access for any given role.

To enable a specific perspective for an Inline Service in Decision Studio, select the perspective in the Decision Center Perspectives folder, then right-click it and select its Properties. You grant access to the perspective for a given role by adding the role to the list of roles that are allowed to use the perspective. For more details, see the section "About Decision Center Perspectives" in *Oracle Real-Time Decisions Platform Developer's Guide*.

5.5 Debugging Role Assignments

As an aid to understanding which Oracle RTD roles have been assigned to a specific user, Oracle RTD provides a resource that is installed as part of Decision Center. If this feature has been enabled, then accessing `http://`*`host`*`:`*`port`*`/ui/hello.jsp` from an HTTP browser will cause Oracle RTD's login page to be displayed. If the user is not already logged in, then a simple page is presented that displays the roles of the user, as in the following example:

- Hello, phil

  You are in these RTD Roles: [RTDAdministrators, RTDUsers]

The roles listed here are a subset of all the roles to which any cluster permissions have assigned in Oracle RTD's Security Manager MBean. The total list of possible Oracle RTD roles is the set that is rendered by the Security Manager MBean operation, **listEveryoneHavingPermissions**.

Note: To avoid any concern that displaying a user's roles might be a privacy issue, `hello.jsp` will not display any roles unless this feature is explicitly enabled by setting the following system property for the JVM, in a manner specific to the application server:

- rtd.showRolesInHello.jsp = true

5.6 Migration Strategy

This section shows the tasks required of existing Oracle RTD customers upgrading from previous releases to 3.0, and contains the following topics:

- Section 5.6.1, "Migrating Oracle RTD Users and Groups to Enterprise Identity Store"
- Section 5.6.2, "Defining Standard Security Roles in JEE Server"
- Section 5.6.3, "Defining Custom Security Roles in JEE Server"
- Section 5.6.4, "Declaring Custom Role References"
- Section 5.6.5, "Assigning Inline Service Permissions to Roles"
- Section 5.6.6, "Assigning Decision Center Perspective Permissions to Roles"

5.6.1 Migrating Oracle RTD Users and Groups to Enterprise Identity Store

The users and groups that had been defined in Oracle RTD's internal identity store should generally be moved to the JEE server's identity store. If the JEE server is integrated into an enterprise-wide identity store, this store will typically already include the users that had previously been entered separately into Oracle RTD's internal identity store.

5.6.2 Defining Standard Security Roles in JEE Server

This process is different for each type of application server, using the vendor's administration console. Examples are shown in the appropriate application server specific chapters.

5.6.3 Defining Custom Security Roles in JEE Server

If the existing installation assigns permissions to users or groups that are more fine-grained than the Oracle RTD's standard roles, then additional enterprise roles must be defined so that permissions can be assigned to them.

For example, if there were user groups named **ILS2-AdminGroup** and **ILS2-UserGroup**, two new roles must be defined, and cluster permissions assigned, as follows:

- **ILS2Developers:** true for users in **ILS2-AdminGroup**
- **ILS2Users:** true for users in **ILS2-UserGroup**

Table 5–8 Cluster Permissions for Custom Roles

Custom Role	Cluster Permissions
ILS2Developers	Deploy Service from Studio Download Service Open Service Open Service for Reading
ILS2Users	Open Service Open Service for Reading

5.6.4 Declaring Custom Role References

For the custom roles defined in the previous section, declare references to each of them in Oracle RTD's deployment descriptors, then redeploy Oracle RTD:

- `RTD.ear` > `ui.war` > `web-inf/web.xml`
- `RTD.ear` > `soap.war` > `web-inf/web.xml`

5.6.5 Assigning Inline Service Permissions to Roles

Use Decision Studio to change any existing Inline Service permission assignments to refer to the JEE Servlet roles managed by the application server, instead of to the user groups previously managed internally by Oracle RTD.

5.6.6 Assigning Decision Center Perspective Permissions to Roles

Use Decision Studio to change any existing Decision Center Perspective permission assignments to refer to the JEE Servlet roles managed by the application server, instead of to the user groups previously managed internally by Oracle RTD.

6

Additional Configuration Settings and Starting Client Tools

This chapter describes additional configuration settings for Decision Center and Real-Time Decision Server, and provides information about accessing Oracle RTD client tools.

This chapter contains the following topics:

- Section 6.1, "Decision Center Browser Configuration"
- Section 6.2, "Real-Time Decision Server Logging Configuration"
- Section 6.3, "Accessing Oracle Real-Time Decisions Client Tools"

6.1 Decision Center Browser Configuration

Decision Center Internet Explorer client browsers should be configured for optimal performance, as follows:

1. In Internet Explorer, choose **Tools** > **Internet Options** to set options.
2. On the **Advanced** tab, deselect **Reuse windows for launching shortcuts**.
3. Ensure that cookies are enabled for the browser.

6.2 Real-Time Decision Server Logging Configuration

Note: In order for localized characters (such as inline service choice or attribute names) to display properly in the log files, these files should have UTF-8 encoding. Add the JVM environment variable `file.encoding=utf8` (for example, `-Dfile.encoding=utf8`) to the JVM configuration either by adding the statement to the server startup script or by invoking the parameters at JVM startup. For more information, see also appendix Appendix A, "System Log and Configuration Files".

The Logger attribute of the **OracleRTD** > **SDManagement** > **SDLogging** MBean is a Log4J-type logger (accessible through JMX clients such as JConsole). For full documentation, see the Log4J website at:

`http://logging.apache.org/log4j`

To change logging configuration parameters:

1. Start Oracle RTD.
2. Start JConsole and log in. See Section 11.3, "Accessing JConsole" for more information.
3. Go to the **Oracle RTD > SDManagement > SDLogging** MBean.
4. Change the logging attributes as necessary:
    - The attributes **Priority** and **InlineServicePriority** can be set to `DEBUG`, `INFO`, `WARN`, or `ERROR` to control the level of logging messages that are written to the server log.
    - The attribute **LogFile** determines the location of the server log and is by default set to `RTD_HOME\log\server.log`.
5. Press **Enter**. To cancel your changes, press the **Esc** key.

6.3 Accessing Oracle Real-Time Decisions Client Tools

Perform the steps in the following sections to start and access the Oracle RTD client tools, such as Decision Studio, Decision Center, and Load Generator. See Section 11.3, "Accessing JConsole" for information about starting the JConsole administration tool.

To access Decision Center, Real-Time Decision Server must be running. Decision Studio and Load Generator can function partially even when Real-Time Decision Server is not running.

This section contains the following topics:

- Section 6.3.1, "Accessing Decision Studio"
- Section 6.3.2, "Accessing Decision Center"
- Section 6.3.3, "Accessing Load Generator"

6.3.1 Accessing Decision Studio

To access Decision Studio, go to the client computer where you installed the Oracle RTD client tools and run `RTD_HOME\eclipse\eclipse.exe`.

6.3.2 Accessing Decision Center

To access Decision Center from any computer, open a Web browser and go to `http://server_name:port/ui`.

The Oracle RTD application port is typically `8080`.

6.3.3 Accessing Load Generator

To access Load Generator, go to the client computer where you installed the Oracle RTD client tools and run `RTD_HOME\scripts\loadgen.cmd`.

7

Production Deployment of Oracle Real-Time Decisions

This chapter explains the various components of an Oracle Real-Time Decisions configuration and introduces the concept of production deployment of Oracle RTD, for a proof of concept/pilot-type implementation or a full production environment.

After an Inline Service is tested and ready for production deployment, you will deploy it to one or more servers for production. The hardware requirements for your system depend on whether you choose a proof of concept/pilot-type configuration or a full production configuration. System configuration may need to be adjusted based upon specific customer requirements and application characteristics.

This chapter contains the following topics:

- Section 7.1, "Oracle Real-Time Decisions Components"
- Section 7.2, "Proof of Concept Configuration"
- Section 7.3, "Full Production Configuration"

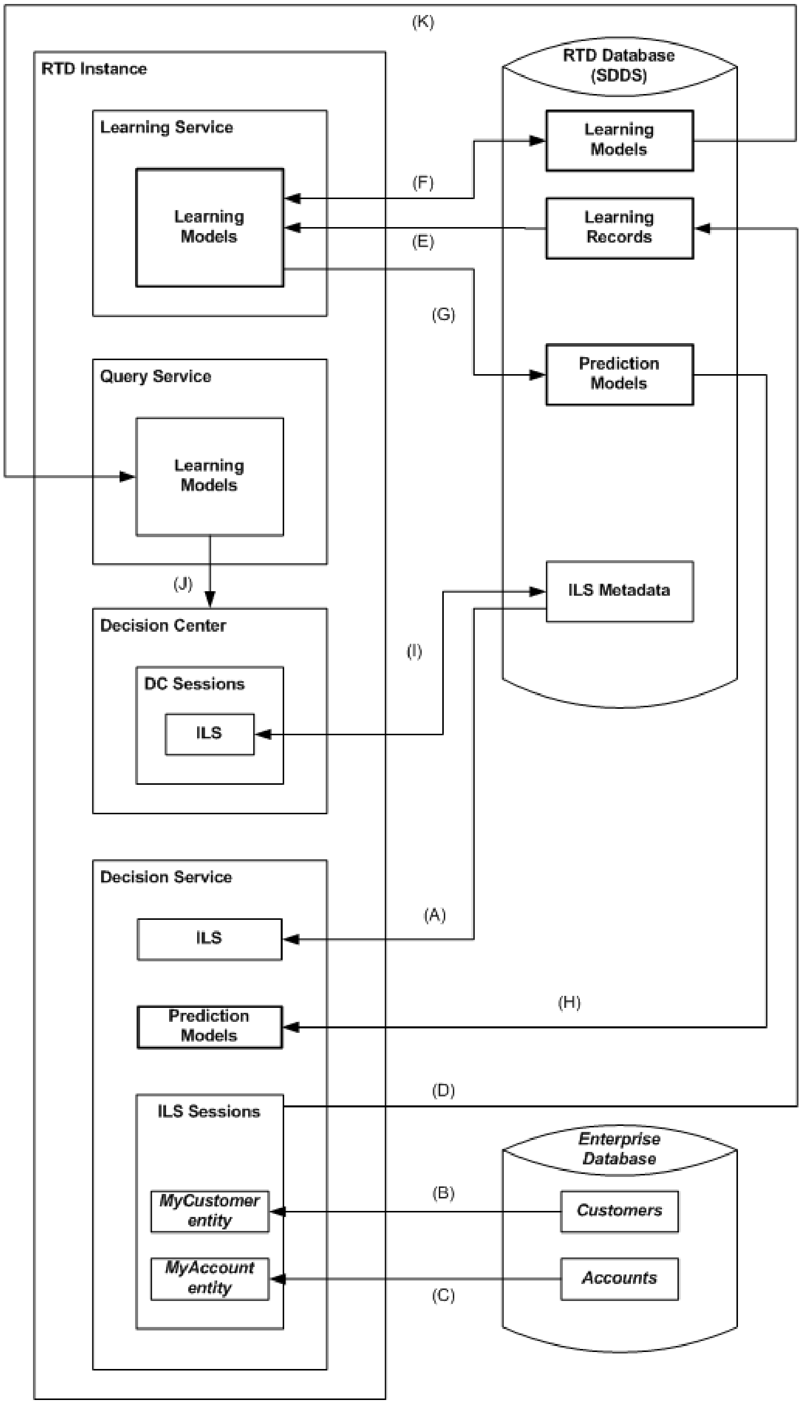
7.1 Oracle Real-Time Decisions Components

Oracle Real-Time Decisions is a real-time decision engine software platform that consists of the following components:

- Real-Time Decision Server
- Decision Studio
- Decision Center
- Load Generator
- Administration (JMX)

Figure 7–1 shows the main service and database components of Oracle Real-Time Decisions and the main runtime interactions between the components. To introduce and illustrate the basic Oracle Real-Time Decisions architecture concepts, the diagram shows a basic Oracle Real-Time Decisions topology, with one occurrence of each of the main services within a single Oracle Real-Time Decisions instance.

Figure 7–1 Single-Instance Oracle Real-Time Decisions Components and Interactions



The main Oracle Real-Time Decisions services are:

- Decision Service

  Decision Service is responsible for tracking customers through Inline Service (ILS) sessions and responding to requests for decisions. It implements the decision logic, computes predictions and scores, and returns sorted results to the calling operational systems. It also creates learning records for model update.

A learning record contains session data and information about events that occurred during the session, for example, the fact that three offers were presented and one was accepted.

- Learning Service

  Learning Service is responsible for updating the models. It reads the learning records created by Decision Service, updates the models and periodically generates new optimized prediction models to be consumed by Decision Service.

- Query Service

  Query Service uses the models produced by Learning Service to provide the data behind Decision Center reports.

- Decision Center

  Decision Center is the web-based application that is responsible for producing the Decision Center reports.

The Oracle RTD database is used to store model data and ILS metadata, in addition to the control and configuration data required for the Oracle Real-Time Decisions environment. An ILS is typically deployed to the database through Decision Studio (not shown in Figure 7–1).

The following provides an overview of the main runtime interactions between Oracle Real-Time Decisions components and Inline Services, and includes references to the processes annotated in Figure 7–1.

- At runtime, ILS metadata is loaded into Decision Service memory - (A). Enterprise data and results of decision requests are stored in ILS session data *(only entity data is highlighted in the diagram, but choice event data is also significant for model updates)* - (B), (C).
- Either during or at the end of ILS sessions, learning records are written out to the database by Decision Service - (D).
- Learning Service reads the learning records, and produces learning models - (E) - which are periodically written out ("persisted") to the database - (F). Learning Service also periodically creates the prediction models - (G) - that are the basis for many Oracle Real-Time Decisions decisions.

  In more detail:

  1. By default, Learning Service wakes up every 15 seconds, and checks to see if there are any new learning records in the database. If so, it retrieves the new records in batches and creates or updates the learning models in Learning Service memory.
  2. Each model has a persistence interval, which controls how often learning models are written to the database. If the persistence interval is reached or exceeded since the last write-out to the database, Learning Service writes out the learning models and deletes the old learning records.
  3. Learning records and learning models contain all the attributes (and their values) defined for the session. The key part of the learning process is selecting those attributes and values that are predictive, that is, which correlate to the output that you want to predict, and generating prediction models from them. In general, prediction models are much smaller than learning models, as only a small proportion of attributes are predictive.
  4. You can select whether or not to generate a prediction model for an Oracle RTD model. If predictive modeling is selected, then another model parameter

controls when to generate the prediction model. This parameter specifies what percentage of new learning records added to the model will prompt the creation of a new prediction model. The new prediction model replaces the old prediction model.

- Prediction models are used by Decision Service as required for Oracle RTD decisions. Decision Service periodically checks if there are any new prediction models, and if so, loads them into its memory - (H). They are retained in Decision Service memory to service subsequent decisions.
- Decision Center loads ILS metadata into Decision Service memory when users log in and select an ILS - (I). As required, Decision Center users may edit the ILS and redeploy it to the database. Decision Center as a reporting tool interacts with Query Service to query the contents of learning models - (J). Query Service itself reads the learning models from the database - (K)..

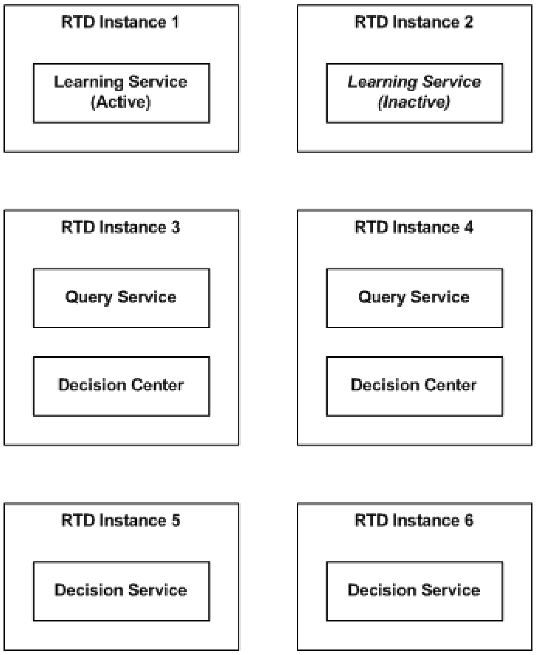
Note: Another Oracle RTD component, Batch Service, does not appear in Figure 7–1. For more information on Batch Service, which is currently co-located with Decision Service, see "Batch Framework Architecture" in *Oracle Real-Time Decisions Platform Developer's Guide*.

7.1.1 Clustering

In many production systems, Oracle RTD is installed and deployed in an Oracle RTD cluster. The following considerations apply to Oracle RTD clusters:

- Learning Service is a cluster singleton, meaning that only one Learning Service instance is active at a time, although it can be deployed to multiple Oracle RTD instances for availability, as in the following possible configuration (Figure 7–2):

Figure 7–2 Learning Service as a Cluster Singleton



- In a cluster, Decision Center and Query Service are typically deployed to one or more dedicated Oracle RTD instances. Decision Center uses Query Service to populate its reports. It is possible to dedicate individual Query Service instances to processing the queries for particular Studies (the collection of models associated with a given Inline Service).

- In a cluster, Decision Service is typically deployed to different Oracle RTD instances than either Learning Service, Query Service or Decision Center, in order to minimize the performance and memory impact of these other components on Decision Service.

7.1.2 Time Windows

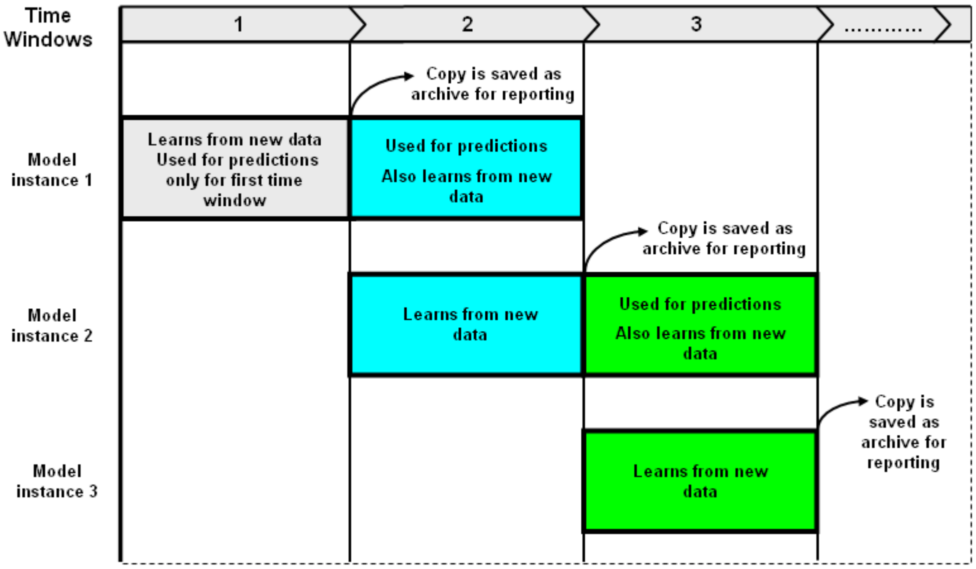
The Time Window parameter influences how models are processed and used. Time Window defines a time period, such as Week, Month, Quarter, Year, which is defined (and can be altered) at the model level.

In addition to the standard modeling processing described in this section, which is independent of time windows, learning models are archived at the end of each Time Window period. These model archives are the basis for Decision Center analytic reports, which show data within and across time periods.

Predictions are always made from the most recent data. In general, they derive from the learning records received so far in the current time window together with all the learning records in the previous time window. During the first time window, predictions are made from what is currently available.

Figure 7–3 shows the model processes as they occur in the initial time windows of a model, and shows how Oracle RTD creates new, overlapping instances of models as time passes.

Figure 7–3 Model Processes in Initial Time Windows


Time Windows
1
2
3
…………
Copy is saved as archive for reporting
Model instance 1
Learns from new data Used for predictions only for first time window
Used for predictions
Also learns from new data
Copy is saved as archive for reporting
Model instance 2
Learns from new data
Used for predictions
Also learns from new data
Copy is saved as archive for reporting
Model instance 3
Learns from new data

Each model instance starts at the beginning of a time window, exists for two time windows, and, apart from the first model instance, overlaps with the previous model instance.

For each model instance, learning information accumulates during its first time window, at the end of which time a copy is saved as archive for reporting. At that point in time:

1. The model instance becomes the "primary model instance", and during its second time window it is used for predictions, while also continuing to learn
2. The next model instance is created, which serves as the "secondary model instance" and is only used for learning purposes until it becomes the primary model itself at the next time window boundary

To summarize:

- Predictions derive from the learning records received so far in the current time window together with all the learning records in the previous time window.
- Apart from the first time window, predictions only derive from a primary model instance, during the second time window of its existence.

Apart from the first time window, the Learning Service holds in memory, for each model, both the primary and secondary model instances, which span the current and previous time windows.

The Query Service loads into memory instances of the models belonging to the studies it has been assigned, for the time windows involved in the queries it receives. This will included archived models if Decision Center users make queries for time windows preceding the current one.

Decision Center reports have a setting to “show incomplete time window”. When checked, reports for the current time window use the secondary model instance and reports for the previous time window use the archived model. When unchecked, reports for the combined current and previous time windows use the primary model instance.

7.2 Proof of Concept Configuration

The intent of a proof of concept/pilot-type configuration is not to support a system in production for an unspecified length of time, but to allow the customer to better understand the benefits of deploying an Inline Intelligence solution. All the Oracle RTD services are deployed on a single server

Table 7–1 lists the hardware requirements for an proof of concept configuration.

Table 7–1 Hardware Requirements for All-In-One Configuration

Hardware	Minimum Configuration
Number of Cores	4
RAM	16 GB

7.3 Full Production Configuration

The Full Production configuration supports a large scale deployment of Oracle RTD. There are a number of factors that determine the system configuration. They include the peak number of requests per second, the number of choices, the number and size of models for the current time window and previous time windows and the number of data sources, among other things.

This configuration involves:

- Multiple Decision Service instance for the processing of integration points
- A Learning Service instance for the processing of the models
- A Backup Service instance for failover shall the primary learning server fail

- One or multiple instances running both Query Service and Decision Center for reporting purpose

The number of Decision Service instances depends on the peak number of requests per second, the number of choices and desired response time. The memory requirements for the Decision Service instances depends on the number of concurrent sessions, size of each session, and the number and size of the prediction models.

The number of cores for the Learning Service instance and backup Learning Service instance depends on how much parallel processing the Learning Service needs to do to keep up with the load generated by the Decision Service instances. The memory requirement of the Learning Service instance and backup Learning Service instance depends on the number and size of models for the current and previous time windows.

The number of Query Service and Decision Center instances depends on either the number of Decision Center concurrent users or the number and size of current, previous and archived models. The more archived models you want to be able to report on, the more memory you will need on your Query Service instance, or have multiple Query Service instances. The number of cores of your Query Service instance(s) depends on the number of Decision Center concurrent users.

Please refer to our published benchmarks for further information on the configuration most appropriate for your business and technical requirements. The Oracle Corporation Professional Services Group can also assist in determining the most appropriate configuration for your needs.

Table 7–2 lists the hardware requirements for a Full Production Configuration.

Table 7–2 Hardware Requirements for Full Deployment Configuration

Server	Component	Minimum Configuration
Decision Service (each instance)	Number of Cores	2
	RAM	4 GB
Learning Service (each, primary or backup instance)	Number of Cores	4
	RAM	16 GB
Query Service and Decision Center(each instance)	Number of Cores	4
	RAM	16 GB
Database Server	Number of Cores	4
	RAM	16 GB

8

Command Line Deployment of Inline Services

In addition to deployment from Decision Studio, Inline Service deployment is also available from the command line. This type of Inline Service deployment can be performed either directly by a command in a command line window or from a script that calls the command. This chapter describes how to set up and use the Oracle RTD command line deployment tool.

This is useful in situations where several developers work on an Inline Service whose files are stored in a source control system. On a scheduled (or on-demand) basis, the Inline Service is retrieved from the source control system onto the production machine or onto another machine that can deploy the Inline Service into a production machine. You can then programmatically deploy the Inline Service to the production RTD server.

As released, the command line deployment tool is located in the zip file, `rtd-deploytool-3.`*`x.x.x.x.x.x.xxxx`*`.zip`, in the directory *`RTD_HOME`*`\client\CommandLineDeploy`.

First, unzip the file `rtd-deploytool-3.`*`x.x.x.x.x.x.xxxx`*`.zip`. The file can be unzipped into any machine running Linux or Windows. The machine must have JDK installed. The JDK version used to run the command line deploy tool must be the same as the JDK version used by the RTD server.

In this section, the directory into which `rtd-deploytool-3.`*`x.x.x.x.x.x.xxxx`*`.zip` is unzipped is referred to as *`RTD_DEPLOYTOOL_DIR`*. The unzipping creates the folder structure `OracleBI/RTDdeploytool` under *`RTD_DEPLOYTOOL_DIR`*. The main command line deployment tool, `deploytool.jar`, which performs the command line deployment, is located in the `RTDdeploytool` folder.

Note: To use the command line deployment tool, users must be associated with an application server role which permits deploying the Inline Service.

For example, if no permission assignments have been removed from the standard Oracle RTD roles as released, then a user in the RTDStudioDeployers role or RTDAdministrators role is permitted to use the command line deployment tool.

8.1 Deploying the Inline Service

To deploy the Inline Service from the command line, perform the following steps:

1. Navigate to the directory containing the `deploytool.jar` file.

   ```
   cd RTD_DEPLOYTOOL_DIR/OracleBI/RTDdeploytool/
   ```

2. Run the following command:

   ```
   java [-Djavax.net.ssl.trustStore="<trust_store_location>"] -jar deploytool.jar
   -deploy [[<named_option>...] source [<positional_option> [...]]
   ```

   The parameters for `deploytool.jar` are as follows:

   - *`<named_option>...`* - one or more of the name-value pairs listed in Table 8–1.

Table 8–1 Command Line Deployment Named Options

Format	Description
`-server` *`address`*	Hostname or IP address for the RTD server. Default is localhost.
`-port` *`number`*	Port on which the RTD server listed. Default is 8080.
`-sslConnection` *`bool`*	Whether SSL is used for connection. Default is false.
`-deploymentState` *`name`*	Deployment state for this Inline Service.Default is Development
`-releaseLock` *`bool`*	Releasing the lock allows other users to edit the Inline Service in Decision Center or to deploy it from Decision Studio. Default is true.
`-terminateSessions` *`bool`*	Terminating active sessions in the Decision Server. Do not use this option when deploying to a production server unless necessary. Default is false.

   - *`source`* - one of the following:
     - Directory: Full path to the Inline Service project folder which holds the Inline Service to be deployed.
     - Zip file: Full path to the zip file containing the Inline Service project folder which holds the Inline Service to be deployed.

       The zip file can contain only one Inline Service with up to one level parent folder.

     Paths that include spaces must be enclosed within double quotes, for example, `"C:\My Projects\CrossSell"`.

   - *`<positional_option>[...]`* - positional parameter list in the format:

     ```
     [username password
     [server[port[ssl[state[unlock[endsessions]]]]]]]
     ```

     as listed in Table 8–2.

Table 8–2 Command Line Deployment Positional Options

Parameter	Description *(includes references to Table 8–1 Format entries)*
`username`	Name of a user logging in to the RTD server.

Table 8–2 (Cont.) Command Line Deployment Positional Options

Parameter	Description *(includes references to Table 8–1 Format entries)*
password	The user's password. For a blank password, specify "".
server	Positional `-server`
port	Positional `-port`
ssl	Positional `-sslConnection`
state	Positional `-deploymentState`
unlock	Positional `-releaseLock`
endsessions	Positional `-terminateSessions`

The command line deployment program deploys Oracle RTD Inline Services found in a source to a server from the command line.

The parameter `-deploy` is the first required argument.

The parameters *source*, *username*, and *password* are mandatory parameters, and are positional. If the values of the *username* and *password* parameters are not specified initially, users will be prompted to enter them.

If values are specified for a named option and its corresponding positional option, then the positional option value overrides the named option value.

Notes:

1. Help text listing and describing the parameters appears if the `jar` file is called without parameters, for example, `java -jar deploytool.jar`.
2. *<trust_store_location>* is the full path to the trust store file. Use the `-Djavax.net.ssl.trustStore` parameter only if `-sslConnection` (or the equivalent positional parameter `ssl`) is set to true.

Examples without named options

```
java -jar deploytool.jar -deploy "c:\my workspace\CrossSell"
scott brighton 192.168.0.15 8080 true Production false
```

```
java -jar deploytool.jar -deploy CrossSell sp34kc slater
```

```
java -jar deploytool.jar -deploy
c:\OracleBI\RTD\examples\CrossSell sdsu b21k7e false QA
```

Examples using named options

```
java -jar deploytool.jar -deploy -server 192.168.0.15 -port 8081
c:\OracleBI\RTD\examples\CrossSell sysman mi22ty
```

```
java -jar deploytool.jar -deploy -port 8081 -server 192.168.0.15
c:\OracleBI\RTD\examples\DC_Demo sonar chimney
```

```
java -jar deploytool.jar -deploy -sslConnection true
CrossSell.zip calzone twostep
```

```
java
-Djavax.net.ssl.trustStore="C:\OracleBI\RTDdeploytool\etc\ssl\sd
trust.store" -jar deploytool.jar -deploy -sslConnection true
-port 8443 "C:\OracleBI\RTD\examples\CrossSell" ssluser psword
```

9

Setting Up and Using Model Snapshots

The Oracle RTD Model Snapshot feature allows you to export data accumulated in Oracle RTD predictive models to external database tables. These results include counts of events, predictiveness values, and correlations. The data exported from the Oracle RTD models can then be analyzed using standard reporting and business intelligence products and techniques.

The data that Oracle RTD collects in its predictive models for a given Inline Service is attached to a Study. The association between an Inline Service and a Study is defined at deployment time.

The Model Snapshot functionality of Oracle RTD operates at the Study level and affects all the models defined in the Inline Service. Using Model Snapshots, you will be able to export the data contained in all the Models of an Inline Service for a given Study.

The data exported by the Model Snapshot feature allows you to replicate and extend the standard choice and choice group level "predictive model" reports provided by Oracle RTD Decision Center. Furthermore, when associated with customer data from a data warehouse, this exported data enables offline customer centric reporting of predictive insights collected and generated by Oracle RTD.

This section consists of the following topics:

- Section 9.1, "Overview of Setting Up and Using Model Snapshots"
- Section 9.2, "Model Snapshot Tables Schema"
- Section 9.3, "Configuring the Model Snapshot Tables"
- Section 9.4, "Populating and Clearing the Model Snapshot Tables"
- Section 9.5, "Creating Reports from the Model Snapshot Data"
- Section 9.6, "Handling Partitions"
- Section 9.7, "Tuning the Model Snapshot Process"

9.1 Overview of Setting Up and Using Model Snapshots

Note: The description of how to set up and use the Model Snapshot functionality of Oracle RTD assumes that you have a running Inline Service with populated predictive models.

There are three main stages in the process of setting up and using model snapshots, as follows:

1. Configuring the model snapshot tables.
2. Populating the model snapshot tables from the learned data, and clearing the tables as required.
3. Creating reports from the model snapshot tables.

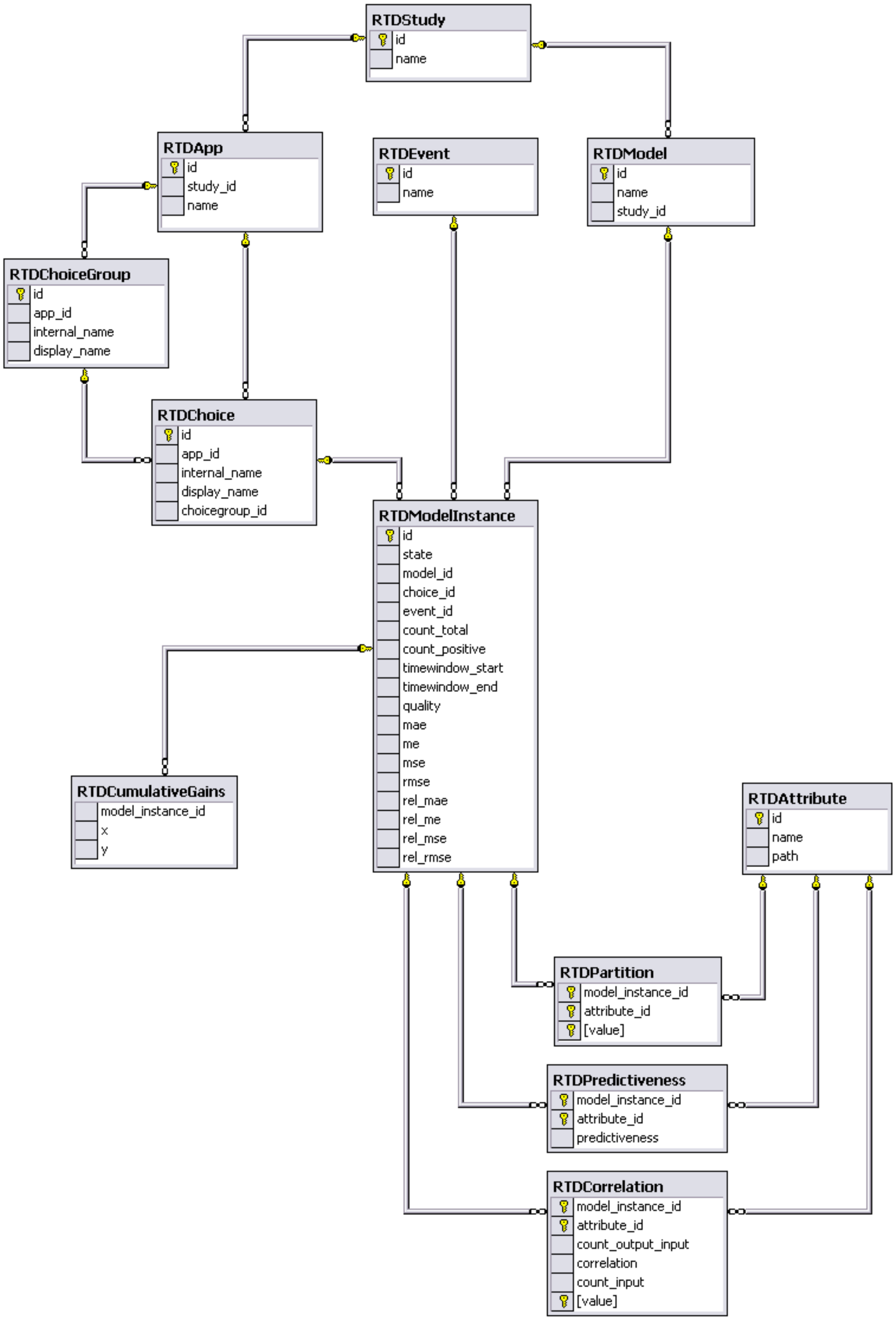
There are two parameters related to model snapshots that are for tuning purposes. For more information, see Section 9.7, "Tuning the Model Snapshot Process."

9.2 Model Snapshot Tables Schema

Oracle RTD exports its predictive model data using a multi-dimensional schema as described in the entity relationship model of Figure 9–1, which shows the model snapshot tables (except for RTDSnapshotSchemaVersion, SDDecisionLog, and SDChoiceEventLog), their columns, and the inter-table relationships, as represented by the connector lines.

Each connector represents a one-to-many relationship, from the table at the top end of the connector to the table at the lower end. In the reverse direction, the relationship from the lower table to the top table is one-to-one.

Figure 9–1 Model Snapshot Tables Schema



The one-to-many relationships between the elements represented by the tables are as follows (each one-to-many relationship implies the corresponding one-to-one relationship in the reverse direction):

Oracle RTD exports its predictive model data using a multi-dimensional schema as described in the entity relationship model of Figure 9–1, which shows the model snapshot tables (except for RTDSnapshotSchemaVersion), their columns, and the inter-table relationships, as represented by the connector lines.

- Each Study can have one or more Applications, and one or more Models.
- Each Application can have one or more Choice Groups, and one or more Choices.
- Each Choice Group can have one or more Choices.
- Each Choice, Event, and Model can have one or more Model Instances.
- Each Model Instance can have one or more Cumulative Gains.

There are also three many-to-many relationships between Model Instances and Attributes, namely Partition, Predictiveness, and Correlation.

The table and column names appear in the following list:

- **RTDApp**

Column	Description
id	Primary key.
study_id	Foreign key to RTDStudy.
name	Application name.

- **RTDAttribute**

Column	Description
id	Primary key.
name	Attribute name.
path	Delimited attribute names showing how this attribute was reached from the root of the session.

- **RTDChoice**

Column	Description
id	Primary key.
app_id	Foreign key to RTDApp.
internal_name	Internal name for Choice.
display_name	Name displayed for Choice.
choicegroup_id	Foreign key to RTDChoiceGroup.

- **RTDChoiceGroup**

Column	Description
id	Primary key.
app_id	Foreign key to RTDApp.
internal_name	Internal name for Choice Group.

Column	Description
display_name	Name displayed for Choice Group.

- **RTDCorrelation**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
attribute_id	Foreign key to RTDAttribute.
count_output_input	Count of those in the population where the value of this output attribute was found given the input.
correlation	Value between -1 and 1. A positive correlation indicates the degree to which the input attribute's value is associated with the value of the output. A negative correlation indicates a negative association.
count_input	Count of those in the population where this value of this input attribute was found.
value	Value for the input attribute.

Note: The sum of over the **RTDCorrelation** column **count_input** for a particular model instance may be less than the related **RTDModelInstance** column **count_total**.

Also, the sum over the **RTDCorrelation** column **count_output_input** for a particular model instance may be less than the related **RTDModelInstance** column **count_positive**.

These results may occur in the following situations:

- Where rows were not stored in **RTDCorrelation** due to exceedingly low correlations, for example, too close to 0.
- Where the **RTDCorrelation** column **value** would have been null, meaning a value for the attribute was absent from the session. In these cases, no correlation is calculated.

- **RTDCumulativeGains**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
x	Cumulative gains curve data point.
y	Cumulative gains curve data point.

- **RTDEvent**

Column	Description
id	Primary key.
name	Event name.

- **RTDModel**

Column	Description
id	Primary key.
name	Model name.
study_id	Foreign key to RTDStudy.

- **RTDModelInstance**

Column	Description
id	Primary key.
state	Model state. Values can be: c: Completed.The model is completed, such as for previous time windows, and is no longer subject to change. It will not be rewritten unless you perform a total dump, or delete the dumps and then perform an incremental dump. d: Combined. The model is for the current and previous time window. s: Split. The model is for the current time window. w: Written. The model is currently being written. Results may be inconsistent.
model_id	Foreign key to RTDModel.
choice_id	Foreign key to RTDChoice.
event_id	Foreign key to RTDEvent.
count_total	Total number of base events.
count_positive	Size of the subset of the population where this non-base event was recorded.
timewindow_start	Start of time window.
timewindow_end	End of time window.
quality	Quality of the model, value vary from 0 to 1. Higher values are better, 0 means nothing was learned.
mae	Mean absolute error.
me	Mean error.
mse	Mean square error.
rmse	Root mean square error.
rel_mae	Relative mean absolute error.
rel_me	Relative mean error.
rel_mse	Relative mean square error.
rel_rmse	Relative root mean square error.

- **RTDPartition**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
attribute_id	Foreign key to RTDAttribute.

Column	Description
value	Attribute value for the partition.

- **RTDPredictiveness**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
attribute_id	Foreign key to RTDAttribute.
predictiveness	Explanatory score of an input attribute for a particular model instance. Values vary from 0 to 1, higher values are better.

- **RTDStudy**

Column	Description
id	Primary key.
name	Study name.

9.3 Configuring the Model Snapshot Tables

The main objectives of this stage are to create the model snapshot tables, and to register them with the application server and the JMX MBeans.

> **Important:** The text values in the model snapshot tables must be case sensitive. If the default setting for your database is case insensitive, make sure that you override the setting when creating the model snapshot tables.

To configure the model snapshot tables:

1. Select the database where your model snapshot tables will be stored.

   > **Note:** You may choose to store the tables in the SDDB database, but this is not recommended for a production system.

2. From the *RTD_HOME*`\scripts` directory, run the command that creates the model snapshot tables:

   ```
   sdexec com.sigmadynamics.tools.SDDBTool.SDDBTool -f -i -I InitSnapshotDb.ctl
   db_type db_host db_port db_name db_runtime_user db_admin_user db_admin_password
   ```

   > **Note:** By replacing the -i parameter with -u, the operation changes from initialization of a new schema to an upgrade of an existing schema.

   The following table describes the parameters for the `sdexec` script:

Parameter	Description
db_type	The database type. Select one of the following: oracle, sqlserver, db2.
db_host	The name of the computer hosting the database server.
db_port	The database port number.
db_name	The name of the database.
db_runtime_user[1]	The user name of the run time user for the system.
db_admin_user	The name of a user that has rights on the database to create tables and stored procedures.
db_admin_password	The password of the administrative user.

[1] For Oracle Database, the *db_runtime_user* and *db_admin_user* are the same user.

3. Create a new Data Source in your application server, that references the database where your model snapshot tables are stored.

   For details of how to create a Data Source in an application server, see Section 3.3.3, "Configuring Additional JDBC Sources" (for WebSphere systems) or Section 4.11, "Configuring New Data Sources" (for WebSphere systems).

   As part of the operation of creating a new Data Source, you provide a new JNDI name, that is used in the steps following.

4. Add the new Data Source to Oracle RTD.

   The actions required to add the new Data Source to Oracle RTD depend on which application server you are using.

   Perform the steps appropriate to your application server, as follows:

   - If your application server is WebSphere, continue at step 5.
   - If your application server is WebLogic, continue at step 8.

   Adding the Data Source to Oracle RTD in WebSphere

5. Open the following file for editing:

   WEBSPHERE_HOME/AppServer/profiles/*profile_name*/config/cells/*server_name*/applications/OracleRTD.ear/deployments/OracleRTD/ls.war/WEB-INF/web.xml

6. Create a new `<resource-ref>` entry, as follows:

   a. Copy the existing `<resource-ref>` entry for the Oracle RTD Database (`SDDS`) and paste it directly below.

   b. Modify the `id` attribute by entering a descriptive value followed by `_LS`, similar to the `SDDS` entry. The `id` must have a unique value within the file.

   c. Modify the `<res-ref-name>` tag by entering the JNDI name (*jndi_name*) that you provided in step 3.

   For example:

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

Save the changes and close the file.

7. Continue at step 10.

Adding the Data Source to Oracle RTD in WebLogic

8. Go to the directory where you expanded the `RTD.ear` file during installation (*`RTD_HOME`*`/package/expanded`).
9. Open the `ls.war` archive, extract `web.xml`, then open `web.xml` for editing. Scroll to the bottom of the file. Copy the section for the definition of the resource reference of `SDDS_LS` and paste it after the existing section. In the copied section, replace the string `SDDS` with the JNDI name (*`jndi_name`*) that you provided in step 3.

   For example:

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

   Save the changes and close the file, then re-archive the file back in `ls.war`.

   Continue at step 10.

10. In JConsole, register the name of the new Data Source in the OracleRTD MBeans:
    - Navigate to MBeans > OracleRTD > SDClusterPropertyManager > Misc.
    - For the **ModelSnapshotDSName** attribute, specify the name of the Data Source that you created for the model snapshots.

9.4 Populating and Clearing the Model Snapshot Tables

You can take snapshots of the model learning data at any time, even if not enough data has been accumulated for prediction purposes.

Each model is defined to have a time window, such as week, month, and quarter. This determines how much data to gather in the learning process, and also influences how much data to write to the model snapshot tables.

You can select the amount of data to write from the following options:

- All the model learning data for a Study
- All the model learning data for a Study for the current time window

You can also delete all model learning data for a Study from the model snapshot tables.

To populate or clear the model snapshot tables:

1. In JConsole, navigate to MBeans > OracleRTD > Learning Server > *your study name*.
2. Click the Operations tab.
3. Click the appropriate snapshot option:
   - **CompleteSnapshot**

Deletes all previously snapped data for the Study, and rewrites all Study data, up to the current time.

- **IncrementalSnapshot**

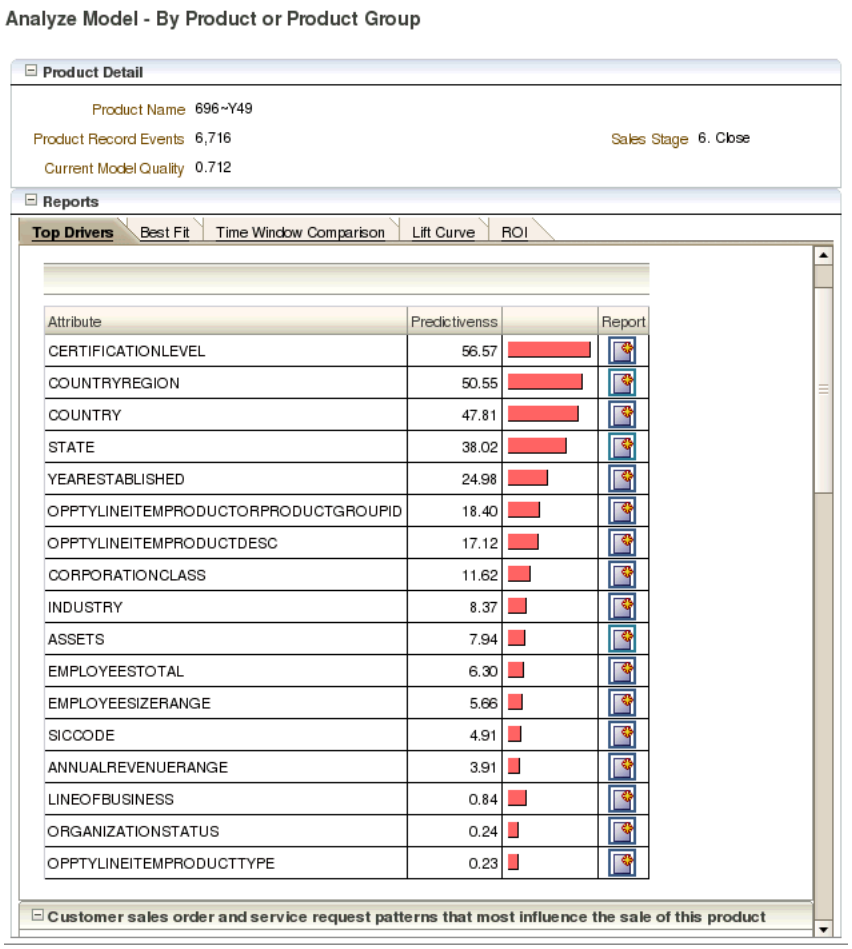
  Deletes Study data for any incomplete time window, and rewrites the current time window's data, up to the current time.

- **DeleteSnapshot**

  Deletes all snapped data for the Study.

9.5 Creating Reports from the Model Snapshot Data

You can create reports from the model snapshot tables, typically by using standard SQL Select commands, or by sending the data to business intelligence products which produce reports similar to the following product sales driver report:



Analyze Model - By Product or Product Group

Product Detail

Product Name 696~Y49
Product Record Events 6,716
Current Model Quality 0.712
Sales Stage 6. Close

Reports

Attribute	Predictivenss		Report
CERTIFICATIONLEVEL	56.57		
COUNTRYREGION	50.55		
COUNTRY	47.81		
STATE	38.02		
YEARESTABLISHED	24.98		
OPPTYLINEITEMPRODUCTORPRODUCTGROUPID	18.40		
OPPTYLINEITEMPRODUCTDESC	17.12		
CORPORATIONCLASS	11.62		
INDUSTRY	8.37		
ASSETS	7.94		
EMPLOYEESTOTAL	6.30		
EMPLOYEESIZERANGE	5.66		
SICCODE	4.91		
ANNUALREVENUERANGE	3.91		
LINEOFBUSINESS	0.84		
ORGANIZATIONSTATUS	0.24		
OPPTYLINEITEMPRODUCTTYPE	0.23		

Customer sales order and service request patterns that most influence the sale of this product

The rest of this section provides examples of scripts that extract information from several of the model snapshot tables. Each script example is followed by sample output. Notes are provided for some of the examples to help you to interpret the results.

The Inline Service used for these examples was a CrossSell application, and the data was generated by running the Oracle RTD Load Generator script to completion, simulating 400,000 user sessions.

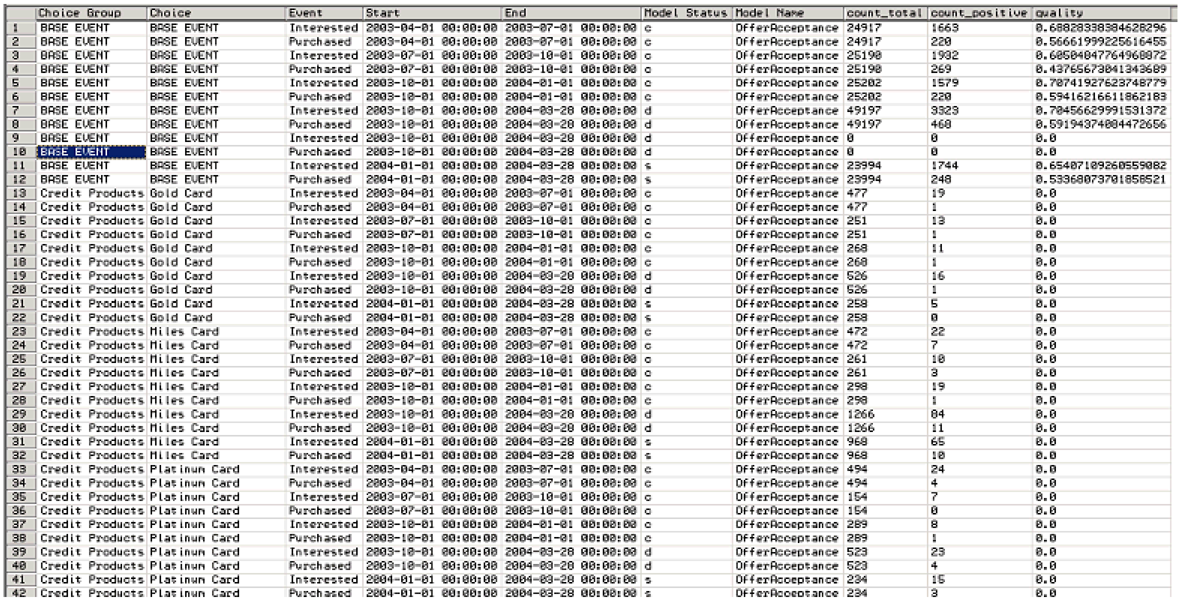
9.5.1 Counts by Choice Query

The following query gets the counts for every Choice, for all time windows:

```
select g.display_name     as 'Choice Group',
       c.display_name     as 'Choice',
       e.name             as 'Event',
      mi.timewindow_start as 'Start',
      mi.timewindow_end   as 'End',
      mi.state            as 'Model Status',
       m.name             as 'Model Name',
      mi.count_total,
      mi.count_positive,
      mi.quality
from RTDApp a
     inner join RTDStudy s          on s.id=a.study_id
     inner join RTDModel m          on m.study_id=s.id
     inner join RTDModelInstance mi on mi.model_id=m.id
     inner join RTDEvent e          on mi.event_id=e.id
     inner join RTDChoice c         on c.id=mi.choice_id
     inner join RTDChoiceGroup g    on c.choicegroup_id=g.id
where a.name='CrossSell'
order by m.name,
        g.display_name,
        c.display_name,
      mi.timewindow_start
```

Figure 9–2 shows the results of the Counts by Choice query.

Figure 9–2 Counts by Choice Query Results



	Choice Group	Choice	Event	Start	End	Model Status	Model Name	count_total	count_positive	quality
1	BASE EVENT	BASE EVENT	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	24917	1663	0.68828338384628296
2	BASE EVENT	BASE EVENT	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	24917	220	0.56661999225616455
3	BASE EVENT	BASE EVENT	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	25198	1932	0.60504047764968872
4	BASE EVENT	BASE EVENT	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	25198	269	0.43765673041343689
5	BASE EVENT	BASE EVENT	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	25202	1579	0.70741927623748779
6	BASE EVENT	BASE EVENT	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	25202	220	0.59416216611862183
7	BASE EVENT	BASE EVENT	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	49197	3323	0.70456629991531372
8	BASE EVENT	BASE EVENT	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	49197	468	0.59194374084472656
9	BASE EVENT	BASE EVENT	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	0	0	0.0
10	BASE EVENT	BASE EVENT	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	0	0	0.0
11	BASE EVENT	BASE EVENT	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	23994	1744	0.65407109260559082
12	BASE EVENT	BASE EVENT	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	23994	248	0.53368073701858521
13	Credit Products	Gold Card	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	477	19	0.0
14	Credit Products	Gold Card	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	477	1	0.0
15	Credit Products	Gold Card	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	251	13	0.0
16	Credit Products	Gold Card	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	251	1	0.0
17	Credit Products	Gold Card	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	268	11	0.0
18	Credit Products	Gold Card	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	268	1	0.0
19	Credit Products	Gold Card	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	526	16	0.0
20	Credit Products	Gold Card	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	526	1	0.0
21	Credit Products	Gold Card	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	258	5	0.0
22	Credit Products	Gold Card	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	258	0	0.0
23	Credit Products	Miles Card	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	472	22	0.0
24	Credit Products	Miles Card	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	472	7	0.0
25	Credit Products	Miles Card	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	261	10	0.0
26	Credit Products	Miles Card	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	261	3	0.0
27	Credit Products	Miles Card	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	298	19	0.0
28	Credit Products	Miles Card	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	298	1	0.0
29	Credit Products	Miles Card	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	1266	84	0.0
30	Credit Products	Miles Card	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	1266	11	0.0
31	Credit Products	Miles Card	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	968	65	0.0
32	Credit Products	Miles Card	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	968	10	0.0
33	Credit Products	Platinum Card	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	494	24	0.0
34	Credit Products	Platinum Card	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	494	4	0.0
35	Credit Products	Platinum Card	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	154	7	0.0
36	Credit Products	Platinum Card	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	154	0	0.0
37	Credit Products	Platinum Card	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	289	8	0.0
38	Credit Products	Platinum Card	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	289	1	0.0
39	Credit Products	Platinum Card	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	523	23	0.0
40	Credit Products	Platinum Card	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	523	4	0.0
41	Credit Products	Platinum Card	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	234	15	0.0
42	Credit Products	Platinum Card	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	234	3	0.0

Notes on the Counts by Choice Query Results

1. In row 13, Choice=Gold Card, Event=Interested, count_total=477, count_positive=19, and quality=0.0. This shows that, out of 477 users that were presented with the Gold Card offer, 19 were interested. The counts are small and the model quality with respect to the Gold Card Choice and the Interested Event is 0.

In row 14, Choice=Gold Card, Event=Purchased, count_total=477, count_positive=1, and quality=0.0. This shows that one user purchased this offer. The model quality with respect to the Gold Card Choice and the Purchased Event is 0.

Both row 13 and row 14 apply to the period of time from April 1, 2003 to July 1, 2003.

2. In the columns Choice Group and Choice, the value BASE EVENT means "in general" or "overall."

   For example, in row 1, Choice=BASE EVENT, Event=Interested, count_total=24917, count_positive=1663, and quality is approximately 0.6882. This means that, in the period between the Start and End dates, a grand total of 24917 users were presented offers, and 1663 of these were interested. The overall model quality for this period of time was about 0.69.

   In row 2, Choice=BASE EVENT, Event=Purchased, count_total=24917, count_positive=220, and quality is approximately 0.5666. This means that, for the same period of time as for row 1, there were 220 Purchased events across all Choices, and the model quality was about 0.57.

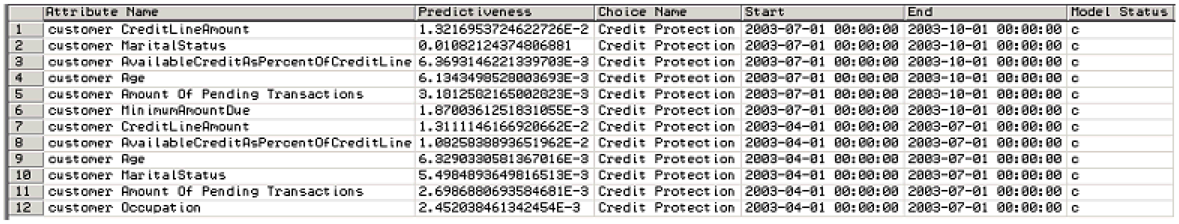
9.5.2 Top Six Predictive Attributes Query

The following query selects the top six predictive attributes, for each time window, for the Credit Protection Choice resulting in the Purchased Event.

```
select a.name                'Attribute Name',
       p.predictiveness      'Predictiveness',
       c.display_name        'Choice Name',
      mi.timewindow_start as 'Start',
      mi.timewindow_end   as 'End',
      mi.state            as 'Model Status'
from RTDApp app
     inner join RTDChoice c         on c.app_id=app.id
     inner join RTDStudy s          on s.id=app.study_id
     inner join RTDModel m          on m.study_id=s.id
     inner join RTDModelInstance mi on mi.model_id=m.id and mi.choice_id=c.id
     inner join RTDEvent e          on mi.event_id=e.id
     inner join RTDPredictiveness p on p.model_instance_id=mi.id
     inner join RTDAttribute a      on a.id=p.attribute_id
where app.name         = 'CrossSell'
  and   c.display_name = 'Credit Protection'
  and   e.name         = 'Purchased'
  and   m.name         = 'OfferAcceptance'
  and   7 > (select count(*)
             from RTDPredictiveness p2
             where p2.model_instance_id = p.model_instance_id
               and p2.predictiveness > p.predictiveness)
order by mi.timewindow_end desc,
          p.predictiveness desc
```

Figure 9–3 shows the results of the Top Six Predictive Attributes query.

Figure 9–3 Top Six Predictive Attributes Query Results



	Attribute Name	Predictiveness	Choice Name	Start	End	Model Status
1	customer CreditLineAmount	1.3216953724622726E-2	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
2	customer MaritalStatus	0.01082124374806881	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
3	customer AvailableCreditAsPercentOfCreditLine	6.3693146221339703E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
4	customer Age	6.1343498528003693E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
5	customer Amount Of Pending Transactions	3.1012502165002023E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
6	customer MinimumAmountDue	1.8700361251831055E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
7	customer CreditLineAmount	1.3111146166920662E-2	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
8	customer AvailableCreditAsPercentOfCreditLine	1.0025838893651962E-2	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
9	customer Age	6.3290330581367016E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
10	customer MaritalStatus	5.4984893649816513E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
11	customer Amount Of Pending Transactions	2.6986880693584681E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
12	customer Occupation	2.452038461342454E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c

9.5.3 Difference Between Expected and Actual Counts Query

The following query shows, for people of different marital statuses, the number who actually purchased credit protection and the number who were expected to do so. The report also shows the difference between the two values, and the importance of the correlation for the offer acceptance.

```
select cor.value               'customer MaritalStatus',
       cor.count_output_input  'Actual Count',
       mi.count_positive*cor.count_input/mi.count_total 'Expected Count',
  100* (  (mi.count_total*cor.count_output_input)
        / (mi.count_positive*cor.count_input) - 1) 'Percent Difference',
       cor.correlation         'Importance',
         c.display_name        'Choice Name',
        mi.timewindow_start as 'Start',
        mi.timewindow_end   as 'End',
        mi.state            as 'Model Status'
from RTDApp app
     inner join RTDChoice c         on c.app_id=app.id
     inner join RTDStudy s          on s.id=app.study_id
     inner join RTDModel m          on m.study_id=s.id
     inner join RTDModelInstance mi on mi.model_id=m.id and mi.choice_id=c.id
     inner join RTDEvent e          on mi.event_id=e.id
     inner join RTDCorrelation cor  on cor.model_instance_id=mi.id
     inner join RTDAttribute a      on a.id = cor.attribute_id
where app.name         = 'CrossSell'
    and c.display_name = 'Credit Protection'
    and e.name         = 'Purchased'
    and m.name         = 'OfferAcceptance'
    and a.name         = 'customer MaritalStatus'
order by mi.timewindow_end desc,
        cor.correlation desc
```

Notes on the Difference Between Expected and Actual Counts Query

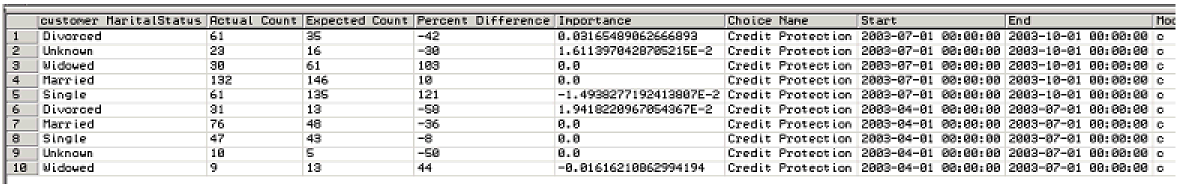
1. The Actual Count, `cor.count_output_input`, is the actual number of people who purchased credit protection, for each marital status.

2. The Expected Count is a simple linear projection of the total count of each marital status, `cor.count_input`, to those that purchased credit protection, as expressed by `mi.count_positive/mi.count_total`.

3. The Percent Difference is 100 * (Actual Count - Expected Count) / Expected Count.

Figure 9–4 shows the results of the Difference Between Expected and Actual Counts query.

Figure 9–4 Difference Between Expected and Actual Counts Query Results



	customer MaritalStatus	Actual Count	Expected Count	Percent Difference	Importance	Choice Name	Start	End	Mo
1	Divorced	61	35	-42	0.03165489062666893	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
2	Unknown	23	16	-30	1.6113970428705215E-2	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
3	Widowed	30	61	103	0.0	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
4	Married	132	146	10	0.0	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
5	Single	61	135	121	-1.4938277192413807E-2	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
6	Divorced	31	13	-58	1.9418220967054367E-2	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
7	Married	76	48	-36	0.0	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
8	Single	47	43	-8	0.0	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
9	Unknown	10	5	-50	0.0	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
10	Widowed	9	13	44	-0.01616210862994194	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c

Notes on the Difference Between Expected and Actual Counts Query Results

1. There are two rows for each marital status, each corresponding to one of the two time periods, April 1, 2003 - July 1, 2003 and July 1, 2003 - October 1, 2003.

9.6 Handling Partitions

The RTDPartition table holds values for partitioning attributes. If a Model is not partitioned, the Model has one Model Instance per time window, and there are no associated rows in the RTDPartition table.

A Model that is split along one or more of its dimensions is a partitioned Model.

As an example, a Model M can be partitioned by two attributes, Marital Status and Favorite Beverage. If there are 3 values for Marital Status (Married, Single, Divorced) and 2 for Favorite Beverage (coffee, tea), then this model has 6 model instances.

In this case, each Model Instance has two associated RTDPartition rows. For example, the Model Instance for the combination (Marital Status=Married and Favorite Beverage=Coffee) would be associated with two RTDPartition rows, containing the following information:

- RTDPartition row 1: Attribute=Marital Status, Value=Married
- RTDPartition row 2: Attribute=Favorite Beverage, Value=Coffee

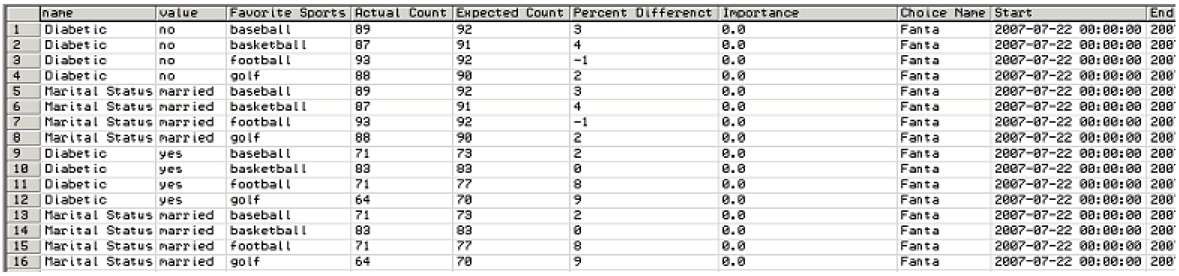
Whether or not a Model is partitioned can influence the results of queries on the model snapshot tables. To avoid repetitions in your results, include RTDPartition and RTDAttribute join conditions in your query.

The following example modifies and extends the Difference Between Expected and Actual Counts query to cover the case of a Model partitioned on two attributes, Diabetic, which has "yes" and "no" values, and Marital Status.

```
select a.name,
       p.value,
      subquery.*
from (select cor.value               'Favorite Sports',
             cor.count_output_input  'Actual Count',
             mi.count_positive*cor.count_input/mi.count_total 'Expected Count',
  100* (  (mi.count_total*cor.count_output_input)
        / (mi.count_positive*cor.count_input) - 1) 'Percent Difference',
       cor.correlation         'Importance',
         c.display_name        'Choice Name',
        mi.timewindow_start as 'Start',
        mi.timewindow_end   as 'End',
        mi.state            as 'Model Status'
        mi.id                   model_instance_id
  from RTDApp app
       inner join RTDChoice c         on c.app_id=app.id
       inner join RTDStudy s          on s.id=app.study_id
       inner join RTDModel m          on m.study_id=s.id
       inner join RTDModelInstance mi on mi.model_id=m.id and mi.choice_id=c.id
       inner join RTDEvent e          on mi.event_id=e.id
       inner join RTDCorrelation cor  on cor.model_instance_id=mi.id
       inner join RTDAttribute a      on a.id = cor.attribute_id
  where app.name       = 'HighlyPartitionedDataset'
    and c.display_name = 'Fanta'
    and e.name = 'loved'
    and m.name = 'SatisfactionModel'
    and a.name = 'Favorite Sports') as subquery
inner join RTDPartition p on subquery.model_instance_id = p.model_instance_id
inner join RTDAttribute a on p.attribute_id = a.id
order by subquery.[End] desc,
         subquery.model_instance_id,
         a.name,
         p.value
         subquery.[Importance] desc
```

Figure 9–5 shows the results of the Partitioned Expected and Actual Counts query.

Figure 9–5 Partitioned Expected and Actual Counts Query Results



	name	value	Favorite Sports	Actual Count	Expected Count	Percent Differenct	Importance	Choice Name	Start	End
1	Diabetic	no	baseball	89	92	3	0.0	Fanta	2007-07-22 00:00:00	200
2	Diabetic	no	basketball	87	91	4	0.0	Fanta	2007-07-22 00:00:00	200
3	Diabetic	no	football	93	92	-1	0.0	Fanta	2007-07-22 00:00:00	200
4	Diabetic	no	golf	88	90	2	0.0	Fanta	2007-07-22 00:00:00	200
5	Marital Status	married	baseball	89	92	3	0.0	Fanta	2007-07-22 00:00:00	200
6	Marital Status	married	basketball	87	91	4	0.0	Fanta	2007-07-22 00:00:00	200
7	Marital Status	married	football	93	92	-1	0.0	Fanta	2007-07-22 00:00:00	200
8	Marital Status	married	golf	88	90	2	0.0	Fanta	2007-07-22 00:00:00	200
9	Diabetic	yes	baseball	71	73	2	0.0	Fanta	2007-07-22 00:00:00	200
10	Diabetic	yes	basketball	83	83	0	0.0	Fanta	2007-07-22 00:00:00	200
11	Diabetic	yes	football	71	77	8	0.0	Fanta	2007-07-22 00:00:00	200
12	Diabetic	yes	golf	64	70	9	0.0	Fanta	2007-07-22 00:00:00	200
13	Marital Status	married	baseball	71	73	2	0.0	Fanta	2007-07-22 00:00:00	200
14	Marital Status	married	basketball	83	83	0	0.0	Fanta	2007-07-22 00:00:00	200
15	Marital Status	married	football	71	77	8	0.0	Fanta	2007-07-22 00:00:00	200
16	Marital Status	married	golf	64	70	9	0.0	Fanta	2007-07-22 00:00:00	200

9.7 Tuning the Model Snapshot Process

You can tune the model snapshot process through the following parameters, available as JMX MBean attributes:

- **ModelSnapshotMinAbsCorrelation** controls whether to dump all correlation rows or to set a minimum correlation value for dumping.
- **ModelSnapshotNumberOfBins** controls the number of bins for model snapshots.

For more information, see Section 11.4.11, "About OracleRTD > Cluster > Query Service."

10

Performance Monitoring

Oracle RTD includes a robust performance monitoring system for observing the behavior of Inline Services. Performance Monitoring parameters are set, and a snapshot view of some of the common counters can be observed, through Fusion Middleware Control. A chronological view can be obtained by enabling the performance monitor. Once enabled, a comma-separated value (CSV) file is produced that can be used to observe behavior over time.

Caution: This file grows without limit, and should be enabled only for active troubleshooting.

This section contains the following topics:

- Section 10.1, "Setting Performance Monitoring Parameters"
- Section 10.2, "Viewing Common Performance Monitoring Snapshot Values"
- Section 10.3, "CSV File Contents"
- Section 10.4, "XLS File Contents"

10.1 Setting Performance Monitoring Parameters

The performance monitoring parameters are set using the SDManagementCluster > Members > Properties > PerformanceMonitoring MBean. You can access this MBean using JConsole; see Chapter 11, "Managing Oracle Real-Time Decisions" for more information.

Table 10–1 describes the properties governing performance monitoring.

Table 10–1 Performance Monitoring Properties

Property Name	Description
DSPerfCounterAppend	If true, performance data is appended to an existing file, if any, otherwise any existing file is overwritten when the server restarts.
DSPerfCounterEnabled	Enables the writing of DS performance counters. This property should not be enabled indefinitely, because the file grows without limit.

Table 10–1 (Cont.) Performance Monitoring Properties

Property Name	Description
DSPerfCounterFileRollingEnabled	If set to TRUE, the filename in DSPerfCounterLogFile is expected to include a date format string, such as `RTD1_ds_perf_%yyyy-MM-dd_hh.csv`. At runtime, the date format string will be replaced with current date and time values. The filename will thus change as the hour changes and a new performance log file begins. The filename MUST include a % if DSPerfCounterFileRollingEnabled is true, indicating the start of the date format string. The date format string can be anything allowed by the Java class SimpleDateFormat (http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html). Note that the date format should not be too granular ; for example, if it includes a minute element, the log file will roll every minute.
DSPerfCounterLogFile	The tab-separated CSV file into which DS performance counts are periodically appended. If MS Excel is available, `ds_perf.xls`, supplied in the `etc` directory of the installation, provides a convenient view. See the first row of `ds_perf.xls` for instructions on linking `ds_perf.xls` to `ds_perf.csv` as a datasource.
DSPerfCounterLogInterval	The update interval in milliseconds for DS performance counts.

10.2 Viewing Common Performance Monitoring Snapshot Values

A snapshot of some of the performance counters is available for viewing through the SDManagementCluster > Members > Decision Service MBean. Press the F5 key to refresh the values.

Performance monitoring does not have to be enabled to use this view.

10.3 CSV File Contents

This section describes the fields of the CSV file containing performance counters.

Table 10–2 Fields of CSV File With Performance Counters

Field Name	Description
Date/Time	The time of day at which the current row of counters was appended to the file. Millisecond precision is available to facilitate correlations with messages in the server's log file.

Table 10–2 (Cont.) Fields of CSV File With Performance Counters

Field Name	Description
Max Allowable Running Requests	The maximum number of Inline Service requests that can run concurrently. The value is derived from configuration settings. It should be chosen to minimize the operating system's thread scheduling overhead, and hence provide maximum throughput for a busy system. The value can be set manually, by setting a non-zero value in either the cluster-wide configuration property, SDManagementCluster > Properties > Misc > IntegrationPointMaxConcurrentJobs, or in the server-specific property, SDManagementCluster > Members > Properties > Misc > IntegrationPointMaxConcurrentJobs. The preferred value is chosen by setting the property to zero, in which case the value is calculated according to the following formula: `NumCPUs * Math.ceil(1/(1-DSRequestIOFactor)) + 5` The formula uses these terms: ■ **NumCPUs:** Server-specific configuration property SDManagementCluster > Members > Properties > Misc > NumCPUs. Use the number of physical CPUs in the machine. ■ **Math.ceil:** Means "round up to the next higher integer value." ■ **DSRequestIOFactor:** Server-specific configuration property SDManagementCluster > Members > Properties > Misc > IntegrationPointRequestIOFactor. The fraction of time Integration Point requests spend doing input/output operations, or otherwise waiting for systems external to this virtual machine. The default value is 0.5.
Peak Requests Running	The largest number of requests that have been running at the same time since the server was started.
Max Requests Running	The largest number of requests that have been running at the same time during the current logging interval.
Requests Running	The number of Inline Service requests that are currently running. This value will always be less than or equal to the Max Allowable Running Requests value.
Request Queue Capacity	The configured maximum number of requests that can wait at the same time in this server to run. This is the value of the cluster-wide property SDManagement-Cluster > Properties > Misc > IntegrationPointQueueSize, or the server-specific property, SDManagement-Cluster > Members > Properties > Misc > IntegrationPointQueueSize. When a request arrives and the request queue is full, the request is rejected and a Server Too Busy error is logged in the server. The property should be set to a value slightly less than the number of concurrent HTTP requests (threads) supported by the Web server; otherwise, the request queue could never fill up, because the requests would be rejected first by the Web server.
Peak Queue Length	The largest number of Inline Service requests that have been waiting at the same time to run in this server since the server started. This will always be less than or equal to **Request Queue Capacity**.

Table 10–2 (Cont.) Fields of CSV File With Performance Counters

Field Name	Description
Max Queue Length	The largest number of Inline Service requests that have been waiting at the same time to run in this server during the current logging interval. This will always be less than or equal to **Request Queue Capacity**.
Requests Waiting (Queue Length)	The number of Inline Service requests that are currently waiting to run.
Requests When Queue Full, Total	The total number of requests that have arrived while the server's request queue was full. Each of these requests was rejected with a Server Too Busy error.
Requests Queued, Total	The total number of Inline Service requests that were required to wait to run until other requests finished running. If all requests are being queued, the system is very busy.
Requests Seen, Total	The total number of Inline Service requests for this server.
Requests In System	The current number of Inline Service requests being processed by this server. The number includes those waiting to run, and those already running.
Timed Out Requests, Total	The total number of requests that have failed to finish running before their guaranteed service level timeout, as specified by cluster-wide property SDManagementCluster > Properties > Misc > IntegrationPointGuaranteedRequestTimeout. This count includes all timed-out requests since the server was started. If this number is growing but the number of queued requests is not growing, this is an indication that the Inline Service logic handling the request is too slow to satisfy the response time guarantee, even on an idle system. One or more Integration Point requests must be optimized, or the response time guarantee must be increased.
Timed Out Requests	The number of requests that failed to finish running before their guaranteed service level.
Timed Out While Running, Total	The total number of requests, observed since the server started, to have started running and not finish within their response time guarantee. The server's processing power consumed by these requests is largely wasted, because the clients will ignore their late responses. When the system is very busy, it sometimes times out requests that are still waiting to run, thus avoiding wasting resources on them.
Timed Out While Running	The number of requests, observed during the current logging interval started, to have started running and not finish within their response time guarantee. The server's processing power consumed by these requests is largely wasted, because the clients will ignore their late responses. When the system is very busy, it sometimes times out requests that are still waiting to run, thus avoiding wasting resources on them.
Timed Out Requests Still Running	The number of requests that have started running, timed out, and are still running. A non-zero value could be an indication of a programming problem in one or more Integration Points.
Request Run Time, Average (ms)	The average time, in milliseconds, during the current logging interval that requests ran. Excludes wait time, if any.

Table 10–2 (Cont.) Fields of CSV File With Performance Counters

Field Name	Description
Request Run Time, Max (ms)	The largest amount of time, in milliseconds, during the current logging interval, that any single request ran. Excludes wait time, if any.
Run Times < [0.1 GRT]	The number of requests that finished running during the current logging interval and ran less than 10% of the configured guaranteed response time. There are nine similarly formatted columns, showing the run time distribution for 0.10, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, and 2.0 times the guaranteed response time.
Run Times < N and >= M	The number of requests that finished running during the current logging interval and ran less than N milliseconds and greater than or equal to M milliseconds.
Run Times >= [2.0 GRT]	The number of requests that finished running during the current logging interval and ran two or more times the configured guaranteed response time.
Request Wait Time, Average (ms)	The average time, in milliseconds, that requests waited on the request queue prior to running or timing out.
Request Wait Time, Max (ms)	The largest amount of time, in milliseconds, during the current logging interval, that any single request waited on the request queue. Includes only those requests that finished running, or timed out before running, during the current logging interval.
Wait Times < [0.1 GRT]	The number of requests that finished running during the current logging interval, and were placed on the request queue before running, but waited there less than 10% of the configured guaranteed response time. There are nine similarly formatted columns, showing the wait time distribution for 0.10, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, and 2.0 times the guaranteed response time.
Wait Times < N and >= M	The number of requests that finished running during the current logging interval and waited on the request queue less than N milliseconds and greater than or equal to M milliseconds before running.
Wait Times >= [2.0 GRT]	The number of requests that finished running during the current logging interval and waited two or more times the configured guaranteed response time before timing out.
Sessions, Current	The number of Decision Server sessions still open in this server.
Sessions, Total	The total number of Decision Server sessions created by this server.
Stale Sessions Closed Asynchronously	The total number of Decision Server sessions that have been closed by kernel jobs, instead of by request threads. This is usually unimportant. In a busy system, most stale sessions are closed by request threads and the kernel jobs are engaged only as the system winds down. It could be of interest to someone observing a lot of kernel-job activity (see **Kernel Jobs Running, Current**).
Stale Sessions Closed by Requests	The total number of Decision Server sessions that have timed out and been closed by request threads. Most sessions will be closed this way, especially on a busy server. After processing an Inline Service request, the calling thread will be asked to close at most one stale session before returning to the caller.

Table 10–2 (Cont.) Fields of CSV File With Performance Counters

Field Name	Description
Requests Forwarded, Current	The total number of Inline Service requests that have been forwarded from this server to other servers, and for which no acknowledgment has yet been received to indicate that the request has been processed by the forwarded-to server.
Requests Forwarded, Peak	Largest number of Inline Service requests forwarded.
Requests Forwarded, Total	Total number of Inline Service requests forwarded.
Received Requests Forwarded, Current	The total number of Inline Service requests that were forwarded from other servers to this server, and which have not yet been completely processed by this server.
Received Requests Forwarded, Peak	Largest number of received Inline Service requests forwarded.
Received Requests Forwarded, Total	Total number of received Inline Service requests forwarded.
Remote Session Keys, Current	The current number of session keys that this server knows reference sessions hosted by other servers. If a request arrives with one of these keys, it will be forwarded to the other server.
Remote Session Keys, Total	The total number of times that session keys were registered in this server for sessions hosted by other servers. This is an aggregation of "Remote Sessions Keys, Current".
Kernel Jobs Running, Current	The number of maintenance activities currently running in the server. Maintenance activities include model maintenance, session timing, and timed-out request processing.
Kernel Jobs Running, Peak	The largest number of maintenance activities that have run at the same time in this server. This value will always be less than or equal to the cluster-wide property SDManagement-Cluster > Properties > Misc > WorkerThreadPoolSize, or the server-specific property, SDManagement-Cluster > Members > Properties > Misc > WorkerThreadPoolSize.
Snapshot Period (ms)	The period of time, in milliseconds, over which the server collected data before logging this row of counters.

10.4 XLS File Contents

This section describes the contents of the Microsoft Excel file, `ds_perf.xls`, included in the `etc` directory of the installation.

At the top, cell B1 contains a comment describing how to link `ds_perf.xls` to the tab-separated counter file as a datasource:

"To specify path to the ds_perf.csv file, place cursor in cell B2 and select "Import External Data" > "Edit Text Import" from the "Data" menu, and navigate to your {$install_directory}\log\ folder and select the ds_perf.csv file. Use default parsing settings when prompted. Data will then be automatically refreshed every 3 minutes. To change interval and other settings, select from the "Data" menu the selection "Import External Data" > "Data Range Properties""

In row 2 are the headers containing the names of each counter. All of the headers from the CSV file appear here, with values below them.

The following columns appear after the values from the CSV file, with formulas showing values calculated from the CSV values:

- **Gross Throughput (req/sec):** The average rate of requests finishing during the current logging interval, in requests per second. The formula is:

  ```
  RequestsFinished / SnapshotPeriod * 1000
  ```

- **Net Throughput (req/sec):** The average rate of requests finishing during the current logging interval, excluding requests that timed out. The formula is:

  ```
  (RequestsFinished - Timeouts) / SnapshotPeriod * 1000
  ```

- **Utilization (%):** The percentage of the server's capacity utilized during the current logging interval. The formula is:

  ```
  (RunTimeAverage * RequestsFinished) / (MaxAllowableRunningRequests *
  SnapshotPeriod) * 100
  ```

  This value can be briefly larger than 100 when requests are finishing that started running in previous logging intervals.

11

Managing Oracle Real-Time Decisions

Oracle RTD uses the J2EE industry standard Java Management Extensions (JMX) to configure and monitor the operation of Oracle RTD. This chapter describes the architecture of Oracle RTD and how to manage Oracle RTD through the Oracle RTD JMX MBeans.

Oracle RTD uses the J2EE industry standard Java Management Extensions (JMX) to configure and monitor the operation of Oracle RTD. You can use the JMX management tool JConsole to manage Oracle RTD. Third-party JMX implementations may also be used to manage Oracle RTD. This document assumes you are using JConsole.

JMX MBeans manage various aspects of Oracle RTD, including logging and Inline Service configuration. They can also be used to assign Oracle RTD specific permissions to security roles and users. Security roles and users are both managed by the J2EE container.

Oracle RTD includes several Services. All of the Oracle RTD services are described later in this section; from the point of view of managing Oracle RTD, the most significant Oracle RTD services are the following:

Decision Service: Runs Inline Services and integrates to enterprise operational processes.

Learning Service: Maintains analytic, self-learning models that underlie Inline Services.

Query Service: Handles Decision Center and Decision Manager model queries.

Deployments of Oracle RTD are often done across multiple servers as well as in clusters to enhance performance in high transaction environments. A relational database is used by each of these Services for retention of code, transactional data and configurations.

This chapter contains the following topics:

- Section 11.1, "Oracle RTD Architecture"
- Section 11.2, "Overview of Oracle RTD and System Properties"
- Section 11.3, "Accessing JConsole"
- Section 11.4, "About JMX MBean Operations and Attributes"
- Section 11.5, "Required System Properties and Other Requirements"

11.1 Oracle RTD Architecture

This section describes the terms and options used for Oracle RTD components and the management of Oracle RTD processes. This section includes the following topics:

- Section 11.1.1, "Definition of an Oracle RTD Cluster"
- Section 11.1.2, "Terminology"
- Section 11.1.3, "Oracle RTD Cluster, Nodes and Services"
- Section 11.1.4, "Java EE Clusters and Their Relationship to Oracle RTD Clusters"
- Section 11.1.5, "Required JVM System Properties for Oracle RTD Cluster Participation"
- Section 11.1.6, "Scale Out"
- Section 11.1.7, "Scale Back and Fail-Over"
- Section 11.1.8, "Overview of Oracle RTD Services"
- Section 11.1.9, "Further Information About Oracle RTD Services"

11.1.1 Definition of an Oracle RTD Cluster

An Oracle RTD cluster is defined as the set of all running RTD.ear instances (Oracle RTD applications) using a common database instance through the data source registered under the JNDI name SDDS. It is a collection of RTD.ear instances, with hub-and-spoke coordination from the database.

11.1.2 Terminology

Node = a running RTD.ear application (also referred to as an Oracle RTD server or Oracle RTD instance)

Service = a class of objects on the cluster providing a given service interface, for example, the Learning Service

Service instance = an instance of a Service type or class

Service acronyms *(used throughout this chapter)*:

- BA = Batch Agent
- BM = Batch Manager
- DS = Decision Service
- LS = Learning Service
- QS = Query Service

Java API = the published Java API for ILS authors unless otherwise specified

Cluster stack = the Java code and data structures implementing Oracle RTD Services and remote procedures calls (RPCs)

Application = an Inline Service (also known as an ILS)

Platform Java = the Oracle RTD system exclusive of Application Java

11.1.3 Oracle RTD Cluster, Nodes and Services

An Oracle RTD cluster has one or more Nodes (JVM processes, each running an RTD.ear application instance).

Each node offers one or more Services, with Service life-cycle managed by the Cluster stack of the node.

11.1.4 Java EE Clusters and Their Relationship to Oracle RTD Clusters

Any number of Java EE clusters may overlap the Oracle RTD cluster, under the following conditions:

- All Oracle RTD cluster nodes receive the same SDDS data source definition
- All Oracle RTD cluster nodes are in the same security realm (Java EE users and roles)

Any Java EE cluster strategy for server availability may be used:

- Passive secondaries may be used provided that the passive secondary has the same JVM system properties as the primary
- Server-down detection and automatic reboot by the Java EE cluster may be used

11.1.5 Required JVM System Properties for Oracle RTD Cluster Participation

Each node in an Oracle RTD cluster must be assigned two invariant JVM system properties, which must be unique to the node across the cluster:

rtd.instanceName - Name of the node. Only the following characters are allowed: alphanumerics, underscore, dash, period.

rtd.HTTPHostAndPort (in the format ***<host>:<port>***) - RPC host and port for this node.

11.1.6 Scale Out

Addition of a node to the Oracle RTD cluster can be done at any time without interrupting existing nodes or Services. The new node must fulfill the following conditions:

- It must face the same database instance over the required SDDS data source.
- It must have the two JVM system properties, rtd.InstanceName and rtd.HTTPHostAndPort, with their required uniqueness, as described in Section 11.1.5, "Required JVM System Properties for Oracle RTD Cluster Participation."

11.1.7 Scale Back and Fail-Over

Removal of a node from the cluster, either through shutdown of the application server or removal of the RTD.ear deployment on an application server, can be performed at any time without interruption to the other nodes.

If the node removed was the active LS and/or BM, another node on which the service is enabled will be activated by the system, ensuring high availability. The same will happen if a node goes down unexpectedly. If no other server has LS or BM enabled, the cluster will have no active LS or BM instance going forward.

11.1.8 Overview of Oracle RTD Services

There are five Oracle RTD services that can be enabled and disabled by Oracle RTD system administrators:

- **Learning Service (LS)** - The Oracle RTD learning service in which all learning and prediction model creation takes place.
- **Decision Service (DS)** - The integrative service facade for data capture and decisioning.

- **Batch Manager (BM)** - The coordinator for batch operations.
- **Batch Agent (BA)** - Both the API entry point to the batch sub-system and the executor for batch jobs.
- **Query Service** - Serves model queries utilizing the currently active and archived models *(functionality that was provided in previous releases by the Learning Service)*.

In addition, there are two implicit Oracle RTD services that are always enabled and active on all nodes:

- **Workbench** (also known as **Management Service**) - the endpoint for ILS deployment from Decision Studio
- **Decision Center** - the web application for the `/ui` reporting web application of Oracle RTD

Enabling and Disabling the Oracle RTD Services

Each node offers five Oracle RTD Services (BA, BM, DS, LS, and QS) that may be enabled or disabled via a JVM system property or JMX (via the Node MBean).

These five Oracle RTD Services are enabled by default on all nodes. Each node also hosts implicit system services that are always available.

An Oracle RTD Service may offer an MBean for configuration of properties of cluster scope, applicable to all Oracle RTD Service instances.

An Oracle RTD Service instance may also register its own MBean locally for performance metric visibility or instance configuration.

Each service instance exists in one of three states: **disabled**, **enabled**, or **active**.

The active state is always set by the system.

A user may define, per node, whether the service is disabled or enabled. If enabled, the system will decide whether to make the service active, as detailed later in this section. Enablement is defined by system property and can be overridden via JMX.

There are two kinds of services:

1. Single-instance

   A single-instance service can only be active on one node at any given time. Servers on which a single-instance service is enabled compete to be the one node where it is active.

2. Multi-instance

   A multi-instance service can be active on any number of nodes in a cluster. Such a service is activated by the system if it is enabled on a node.

The five standard Oracle RTD Services are classified as either single-instance or multi-instance as follows:

- Learning Service (LS): Single-instance service
- Decision Service (DS): Multi-instance
- Batch Manager (BM): Single-instance
- Batch Agent (BA): Multi-instance
- Query Service (QS): Multi-instance

Notes on the Extent of Oracle RTD Service Enablement

Learning Service (LS) - There exists at most one active instance in the cluster. Enablement is defined by a system property and JMX. Servers on which the LS is enabled compete to be the active LS. Only one can be active at any point in time.

Decision Service (DS) - Any number of active instances may exist on the cluster (one instance per node). Enablement is defined by a system property and JMX. This service is made active (by the system) if it is enabled on a node.

Batch Manager (BM) - There exists at most one active instance in the cluster. Enablement is defined by system property and JMX. Servers on which the BM is enabled compete to be the active BM singleton. Only one can be active at any point in time.

Batch Agent (BA) - Any number of active instances may exist on the cluster (one instance per node). Enablement is defined by a system property and JMX. This service is made active (by the system) if it is enabled on a node.

Query Service (QS) - Any number of active instances may exist on the cluster. Enablement is defined by a system property and JMX. This service is made active by the system if it is enabled on a node.

[Note that the implicit services WorkBench and Decision Center are always active on all nodes.]

Service Dependencies

All services can be independently enabled or disabled on any node, using JMX or system properties.

However, both the DS and the Workbench services require BA enablement if an ILS with a batch job definition is deployed.

> **Note:** If you want to create and deploy an ILS with a batch job, ensure that the BA is enabled on all DS nodes (and any other node to which the ILS's are deployed).

11.1.9 Further Information About Oracle RTD Services

This section provides more details about some of the Oracle RTD Services:

- Learning Service Details
- Query Service Details

11.1.9.1 Learning Service Details

Learning throughput of the learning records is now increased through the implementation of the following:

- Parallel model learning

  When learning occurs at the end of an integration point request, or at the close of the Oracle RTD session, such that more than one model is to be updated by the learning, the updates to the models occur in parallel.

- Read-ahead caching of learning records

  The learning server maintains an active read-ahead queue of learning records to incorporate into its learning models. It is continuously updating this queue of records from the database, in order to offer the learning algorithms the next record as fast as possible.

- Asynchronous deletion of processed learning records

  After learning has occurred on learning records, those learning records are expunged from the database in an asynchronous manner that prevents any decrease in learning throughput in the learning server.

11.1.9.1.1 Configuration of the Learning Server

The Java system properties listed in this section direct the behavior of the learning server. They all have default values, and typically you do not need to modify these values. If you want to change any values, you must set the new values on the Learning Server JVM before starting the JVM.

1. **rtd.numCPUs** - to specify the number of threads to use in the parallel learning thread pool. This is recommended to be less than or equal to the number of CPU cores on the machine running the LS instance. Default: The LS instance detects the number of CPU cores on its machine and defaults the thread count to this number. You do not typically need to modify this setting.

For hardware with a very large number of CPU cores, careful testing may determine that a number smaller than the maximum number of CPU cores may achieve higher throughput.

2. **rtd.LearningDispatcher.LearningDurationMS** - duration of the recurring main task, during which learning records are processed. Default is 15000 (15 s).

You do not typically need to modify this setting.

3. **rtd.LearningDispatcher.LearningSleepMS** - sleep time between executions of the recurring main loop task. Default is 100 (100m s).

You do not typically need to modify this setting.

4. **rtd.LearningDispatcher.ProcessingSize** - number of Learning Records to process as a group before checking for timeout in the "Learn Learning Records" sub-task of the main loop task. Default is 20. The LS will process this many Learning Records before checking whether the timeout set in #2has been met.

You do not typically need to modify this setting.

5. **rtd.LRQueueCapacity** - the capacity of the LS queue the pre-reads Learning Records from the database (from the SDLearning table). Default is 10,000.

Recommendation: set to ~5% of expected learning records generated per hour. That is, if the system is expected to generate 1,000,000 learning records/hour, set the Queue capacity to 50,000.

6. **rtd.LRQueueRefillRatio** - Refill Ratio of the LS queue - this is the ratio of queue size/capacity that induces launch of a new reader task when size is reduced by 1 in Learning Record processing. Default is 0.5 (i.e. when polling removes a Learning Record such that size < ½ capacity, a reader task is launched if one is not already active).

You do not typically need to modify this setting.

7. **rtd.LRQueueSleepIntervalSec** - Sleep time, in seconds, for the LS queue reading task. If the reading task encounters an empty SDLearning table, it sets a sleep time for the next launch of itself - on the next launch, it sleeps for this time before resuming. Default is 10 sec.

You do not typically need to modify this setting.

11.1.9.2 Query Service Details

Studies can be assigned to QS instances via JMX (and *not* through a JVM system property). One study can be assigned to more than one instance. This is to enable Oracle RTD systems with asymmetric study weights (that is, with one or a few very large and important studies and a number of smaller or less important studies) to dedicate appropriately sized hardware to serving queries for large studies.

Note that the setting of study assignments only through JMX prevents the need to restart any server in the QS array. Study assignment is a cluster-wide setting - that makes it possible to manage all assignments throughout the cluster from a single node (any node in the QS array). The assignment persists beyond the start and stop of each QS instance.

A QS instance is designated a "universal processor" if it has no specific study assignment. In this case it can potentially handle queries for any study. A QS instance that is explicitly assigned a study is called a "dedicated processor."

The preference for processing a query related to a given study is given to a dedicated QS instance that is explicitly assigned that study.

The cluster maintains a partial mapping of studies to the instances that should process queries pertaining to those studies, and monitors which is the "current processor" for a a study.

There is a strong preference for "dedicated over universal" processing: if a universal processor is currently handling queries for a study (because no dedicated processor was previously available), and a dedicated processor comes online, all subsequent queries for that study are directed to the dedicated processor. If several processors dedicated to that study become available, the QS instance with the highest available memory among the dedicated processors for the study will handle the query.

When there is no current processor for the study, it will be processed by the QS node with the highest available memory among the dedicated processors for that study; if no dedicated processors for it exist a universal processor with the highest amount of available memory will be selected to handle the query. The handler of the query will be designated as the current processor for the query.

Upon receipt of a collection of queries, a QS instance determines, for each query, which QS instance should handle that query, as described previously. If the determined QS instance happens to be the current instance, no forwarding occurs. In theory, a collection of queries can therefore be dispatched to multiple QS instances. In practice, however, a collection of queries typically will pertain to one study and therefore be handled by a single QS instance, local or remote.

The assignment of studies to QS instances is achieved through the JMX Query Service MBean, which exposes the list of all QS instances in the cluster. System administrators can add a comma-separated list of study names beside each QS instance name to make study assignments. The keyword "*" indicates that the QS instance can service all queries. "*" is the default for all QS instances. For more information, see Section 11.4.11, "About OracleRTD > Cluster > Query Service."

11.2 Overview of Oracle RTD and System Properties

All configuration in Oracle RTD is exposed in MBeans.

Types of properties

- Cluster-scoped properties pertain to the cluster as a whole. They apply to all nodes. Each node sees and uses the same value for each cluster-scoped property.

- Cluster-scoped properties which are locally overrideable have cluster-level defaults that apply to each node unless they are overridden on that node.
- Node-scoped properties apply to a given node only and do not have a cluster-wide default.
- Node-scoped immutable properties are set as system properties and may be exposed as read-only MBean properties.

Any property that may be defined locally for a node may be defined by JVM system property. All RTD JVM system properties now conform to the syntax "rtd.X", with the name following the "rtd." prefix. Any property set by JVM system property is invariant and node-scoped, with 1 exception - service enablement (see below)

Overrides through JMX are persisted and thereby maintain their values through node restarts.

Service Enablement

All 5 services (DS, LS, BA, BM, QS) are enabled by default on all nodes in the cluster.

Any service may be disabled at node boot by setting the JVM system property "**rtd.<*name*>Enabled=false**" where name is the (concatenated) name of the service *("LearningService", "BatchAgent", "BatchManager", "DecisionService", or "QueryService").*

The JVM system property defines the service enablement state at node boot only. This state may be updated via JMX at any time. For example, a node on which the DS was disabled can have the DS enabled via JMX (in the Node MBean, see Section 11.4.12, "About OracleRTD > Cluster > [Node].").

This provides for service migration across the cluster without node restarts. Service enablement state is not persisted - at node reboot, the service is again enabled by default, unless overridden with a JVM system property.

11.3 Accessing JConsole

The steps to access JConsole depend on which application server you are using.

This section contains the following topics:

- Section 11.3.1, "Accessing JConsole for WebLogic"
- Section 11.3.2, "Accessing JConsole for WebSphere"

11.3.1 Accessing JConsole for WebLogic

To access JConsole for WebLogic:

1. Run `JDK_HOME/bin/jconsole`.

   For example:

   ```
   C:\Program Files\Java\jdk1.7\bin\jconsole.exe
   ```

   If you have more than one JDK on your computer, it does not matter which *JDK_HOME* you use. You can run JConsole from any JDK.

   > **Note:** If you are running JConsole on UNIX, you must first add *JAVA_HOME*`/bin` to the beginning of the `PATH` operating system environment variable.

2. Select **Remote Process**.
3. In the Remote Process box, enter *(using the format <hostname>:<port>)* - the name of the host where Real-Time Decision Server is running and the JMX remote port number.
4. For **User Name** and **Password**, enter the credentials you specified when you set up JConsole. See Section 3.1.5, "Configuring Oracle RTD Security in WebLogic" for more information.

11.3.2 Accessing JConsole for WebSphere

To access JConsole for WebSphere:

1. On the computer where WebSphere is installed, run the batch script to start JConsole. See Section 3.12.2, "Setting Up a Batch File for JConsole" for more information.
2. Log in as the JConsole user you created when you set up JConsole (for example, `jmx_admin`).

Note: When Oracle RTD is running on WebSphere, JConsole must be run on the same computer as the WebSphere server, by default. This requirement is due to WebSphere limitations.

11.4 About JMX MBean Operations and Attributes

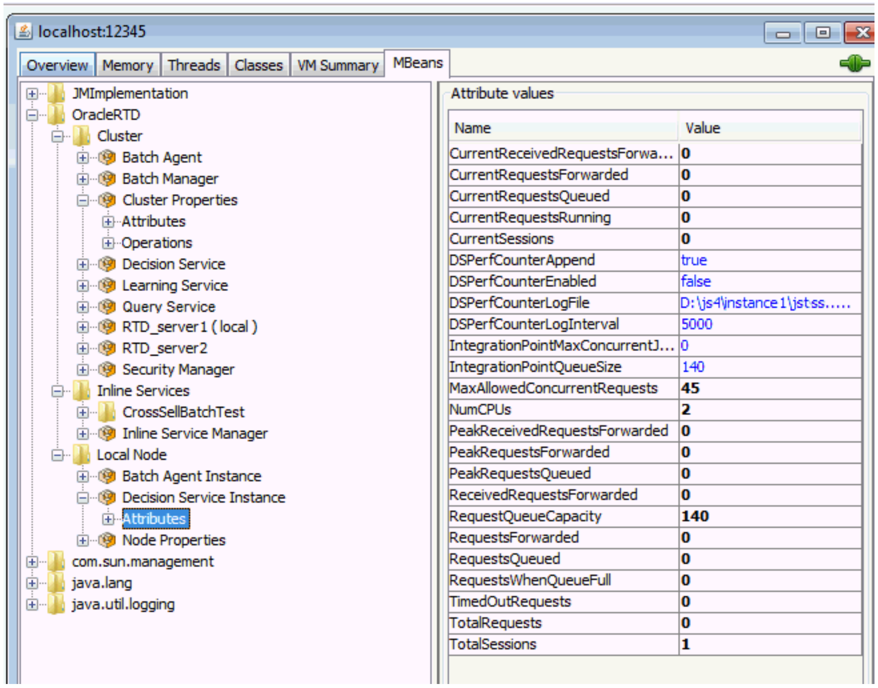
Oracle RTD MBeans can be accessed through the **OracleRTD** folder in the JConsole **MBeans** tab. These MBeans can be used to manage various aspects of Oracle RTD. Each MBean consists of attributes and operations that can be used for informational and administration purposes.

The OracleRTD MBeans break down into the following top-level categories:

- Cluster Category
- Inline Services Category
- Learning Server Category
- Local Node Category
- Study Category

Figure 11–1 shows an example of the OracleRTD MBeans as they appear in JConsole *(the specific attributes displayed there are for the Decision Service Instance MBean, which appears under the Local Node category).*

Figure 11–1 Example of OracleRTD MBeans



Name	Value
CurrentReceivedRequestsForwa...	**0**
CurrentRequestsForwarded	**0**
CurrentRequestsQueued	**0**
CurrentRequestsRunning	**0**
CurrentSessions	**0**
DSPerfCounterAppend	true
DSPerfCounterEnabled	false
DSPerfCounterLogFile	D:\js4\instance1\jstss.....
DSPerfCounterLogInterval	5000
IntegrationPointMaxConcurrentJ...	0
IntegrationPointQueueSize	140
MaxAllowedConcurrentRequests	**45**
NumCPUs	**2**
PeakReceivedRequestsForwarded	**0**
PeakRequestsForwarded	**0**
PeakRequestsQueued	**0**
ReceivedRequestsForwarded	**0**
RequestQueueCapacity	**140**
RequestsForwarded	**0**
RequestsQueued	**0**
RequestsWhenQueueFull	**0**
TimedOutRequests	**0**
TotalRequests	**0**
TotalSessions	**1**

11.4.1 Cluster Category

The Cluster category contains:

- The Service type MBeans: Batch Agent, Batch Manager, Learning Service, Decision Service, and Query Service - described in later sections.
- The Cluster Properties MBean, providing access to all cluster-scoped non-service properties.
- The Security Manager MBean, providing cluster-wide security configuration.
- One MBean for each node in the cluster, termed a "Node" MBean. Shown in Figure 11–1 are two: **RTD_server1** and **RTD_server2**.

11.4.2 Inline Services Category

The Inline Services category contains:

- The Inline Service Manager, providing a cluster-wide view of the deployed ILS's on the cluster.
- One MBean for each deployed ILS. Figure 11–1 shows the MBean for the CrossSellBatchTest ILS.

11.4.3 Learning Server Category

The Learning Server category contains:

- One MBean for each study, which displays a list of the populated models for the study.

11.4.4 Local Node Category

The Local Node category contains:

- The Batch Agent Instance MBean, if the BA is running on the node. This MBean provides access to instance-specific properties.
- The Decision Service Instance MBean, if the DS is running on the node. Figure 11–1 shows an example of a Decision Service Instance MBean. The properties in blue font are editable.
- The Node Properties bean, providing access to node-scoped non-service-instance properties.

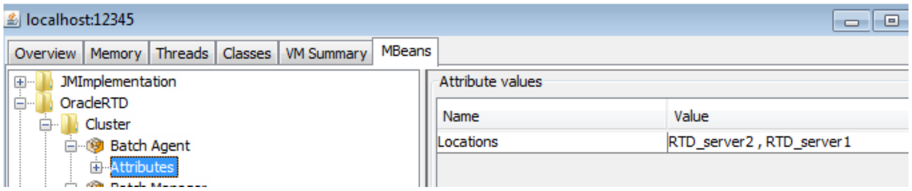
11.4.5 Study Category

The Study category contains:

- One MBean for each populated model in each study.

11.4.6 About OracleRTD > Cluster > Batch Agent

The Batch Agent service MBean reveals the locations of the running Batch Agent instances.



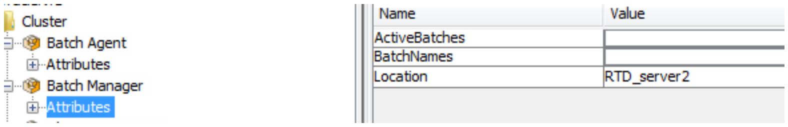
Name	Value
Locations	RTD_server2 , RTD_server1

The Cluster > Batch Agent MBean has the following attributes:

Attribute	**Description**
Locations	Locations of the running Batch Agent instances.

11.4.7 About OracleRTD > Cluster > Batch Manager

The Batch Manager service MBean displays all batch names, all active batch jobs in the cluster, and the location of the Batch Manager singleton instance.



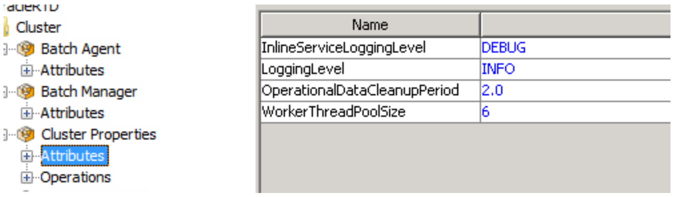
Name	Value
ActiveBatches	
BatchNames	
Location	RTD_server2

The Cluster > Batch Manager MBean has the following attributes:

Attribute	**Description**
ActiveBatches	List of brief status information for all batch jobs currently running, paused, or waiting to run. The list could be empty.
BatchNames	List of batches registered with the batch framework.
Location	Location of the Batch Manager singleton instance.

11.4.8 About OracleRTD > Cluster > Cluster Properties

The Cluster Properties MBean provides access to the following updateable cluster-wide properties:



Name	
InlineServiceLoggingLevel	DEBUG
LoggingLevel	INFO
OperationalDataCleanupPeriod	2.0
WorkerThreadPoolSize	6

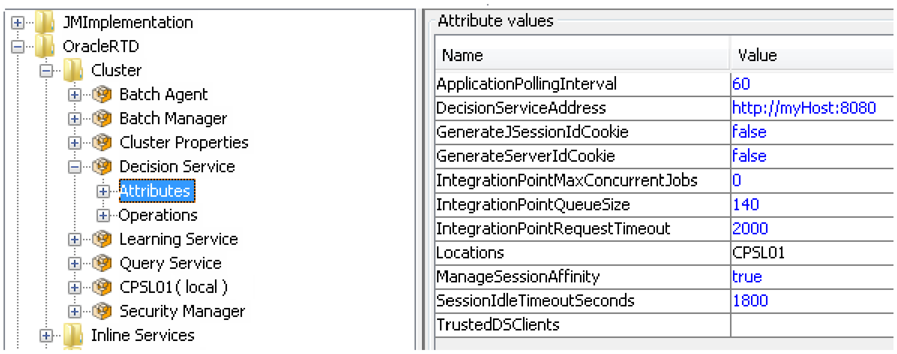
The Cluster > Cluster Properties MBean has the following attributes:

Attribute	**Description**
LoggingLevel	The cluster-wide logging level for Oracle RTD platform code. Default: INFO. This is a cluster-scoped property, set in the Cluster Properties MBean, which may also be overridden at the local level using the Node Properties MBean or JVM system property "rtd.LoggingLevel." This is a dynamic re-configuration property - restart of the cluster or local node is not required. If updated, the new value must be one of ERROR, DEBUG, WARN or INFO, ignoring case. Any other input value is ignored. *In previous releases, this was known as SDLogging.Priority.*
InlineServiceLoggingLevel	The cluster-wide logging level for logging performed by an ILS using the function logDebug(), logInfo(), and so on. Default: INFO. This is a cluster-scoped property, set in the Cluster Properties MBean, which may also be overridden at the local level using the Node Properties MBean or JVM system property "rtd.InlineServiceLoggingLevel." This is a dynamic re-configuration property - restart of the cluster or local node is not required. If updated, the new value must be one of ERROR, DEBUG, WARN or INFO, ignoring case. Any other input value is ignored. *In previous releases, this was known as SDLogging.InlineServicePriority.*
OperationalDataCleanupPeriod	The frequency, in hours (decimals are accepted), at which the system cleans up database tables containing operational data: choice history, statistics. Default is 2.0. A cluster-scoped property, not overridden at the local level. This is a dynamic re-configuration property - restart of the cluster is not required.
WorkerThreadPoolSize	The thread pool size for system tasks (not for normal Integration Point request processing). This is a cluster-scoped property, set in the Cluster Properties MBean, which may also be overridden at the local level using the Node Properties MBean or JVM system property "rtd.WorkerThreadPoolSize." Default: 6. This is not a dynamic re-configuration property - restart of the cluster or local node is required.

The Operations include a **restoreDefaultProperties()** operation to restore the system defaults for these properties.

11.4.9 About OracleRTD > Cluster > Decision Service

The Decision Service MBean provides access to the following properties:



Name	Value
ApplicationPollingInterval	60
DecisionServiceAddress	http://myHost:8080
GenerateJSessionIdCookie	false
GenerateServerIdCookie	false
IntegrationPointMaxConcurrentJobs	0
IntegrationPointQueueSize	140
IntegrationPointRequestTimeout	2000
Locations	CPSL01
ManageSessionAffinity	true
SessionIdleTimeoutSeconds	1800
TrustedDSClients	

The Cluster > Decision Service MBean has the following attributes:

Attribute	Description
ApplicationPollingInterval	The frequency, in seconds, that DS polls for new application deployments. A cluster-scoped property, not overridden at the local level. Updateable in the Decision Service MBean. Default: 60. This is a dynamic reconfiguration property - restart of the cluster is not required.
DecisionServiceAddress	The proxy or direct DS node address to which DC directs calls. A cluster-scoped property, not overridden at the local level. Updateable in the Decision ServiceMBean. Default: http://myHost:8080. This is a dynamic reconfiguration property - restart of the cluster is not required.
GenerateJSessionIdCookie	Set to true to have Decision Service associate Integration Point requests with HTTP sessions, thus causing the Web container to generate container-specific session-affinity cookies. Cluster-scoped. No local override. Default: false.
GenerateServerIdCookie	Set to true to have Decision Service supply a cookie named ORA_RTD_DS_ServerID identifying the DS session serving the current Integration Point request. Cluster-scoped. No local override. Default: false.

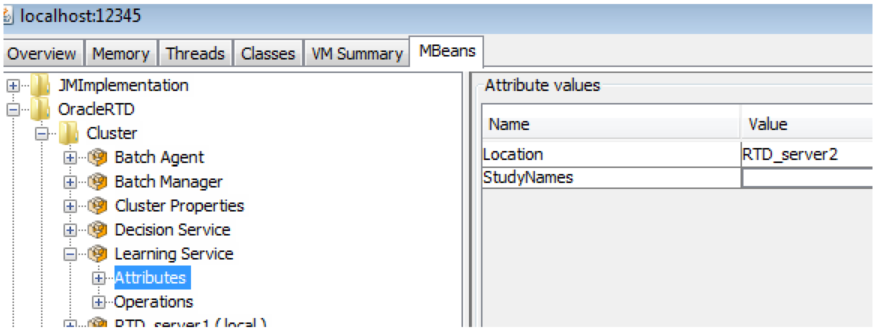
Attribute	Description
IntegrationPointMaxConcurrentJobs	The maximum number of concurrent integration point requests in any DS node in the cluster. A cluster-scoped property, set in the Decision Service MBean, which may also be overridden at the local level using the Decision Service Instance MBean or JVM system property "rtd.IntegrationPointMaxConcurrentJobs." Default: 0, indicating no maximum. This is a dynamic reconfiguration property - restart of the cluster or local node is not required.
IntegrationPointQueueSize	The request wait queue size in the DS. A cluster-scoped property, set in the Decision Service MBean, which may also be overridden at the local level using the Decision Service Instance MBean or JVM system property "rtd. IntegrationPointQueueSize". Default: 140. This is a dynamic reconfiguration property - restart of the cluster or local node is not required.
IntegrationPointRequestTimeout	The time out period, in mSec, for an integration point request into any DS node in the cluster. A cluster-scoped property, not overridden at the local level. Updateable in the Decision Service MBean. Default: 2000. This is a dynamic reconfiguration property - restart of the cluster or local node is not required. In previous releases, this was known as IntegrationPointGuaranteedRequestTimeout.
Locations	The locations of the running DS instances in the cluster.
ManageSessionAffinity	Whether the cluster manages session affinity with DS request forwarding. Default is TRUE. A cluster-scoped property, not overridden at the local level. Updateable in the Decision Service MBean. This is a dynamic reconfiguration property - restart of the cluster is not required. In previous releases, this was known as DSManagesSessionAffinity.
RestrictDSClients	Controls whether the hosts that can send DS requests are restricted to a fixed list of trusted IP addresses. If false, all clients are allowed. If true (the default), localhost is implicitly allowed, plus those clients whose IP addresses appear in TrustedDSClients (a semicolon delimited list). By default,TrustedDSClients is empty, that is, only localhost is allowed as a client. Default: True. See also TrustedDSClients.

Attribute	Description
SessionIdleTimeoutSeconds	The period, in seconds, after which sessions time out with inactivity. A cluster-scoped property, not overridden at the local level. Updateable in the Decision Service MBean. Default: 1800. This is a dynamic reconfiguration property - restart of the cluster is not required. In previous releases, the equivalent attribute was known as DSSessionIdleTimeoutMillisec, with value in mSec.
TrustedDSClients	The IP address list (semi-colon separated) of trusted DS clients. localhost is always trusted. A cluster-scoped property, not overridden at the local level. Updateable in the Decision Service MBean. Default: null (no trusted clients other than localhost). This is a dynamic re-configuration property - restart of the cluster is not required. See also RestrictDSClients.

The Operations include a **restoreDefaultProperties()** operation to restore the system defaults for these properties.

11.4.10 About OracleRTD > Cluster > Learning Service

The Learning Service MBean provides access to the following properties:



Name	Value
Location	RTD_server2
StudyNames	

The Cluster > Learning Service MBean has the following attributes:

Attribute	Description
Location	The location of the running LS instance in the cluster.
StudyNames	A list of all the actively learning studies running on the Learning Server. *Note: You can view the models of a study by clicking on the study under OracleRTD> Study > [Study Name].*

The Cluster > Learning Service MBean has the following operations:

deleteStudy()

Deletes this study.

StudyName The name of the study.

saveStudy()

Saves this study.

StudyName The name of the study.

saveAllStudies()

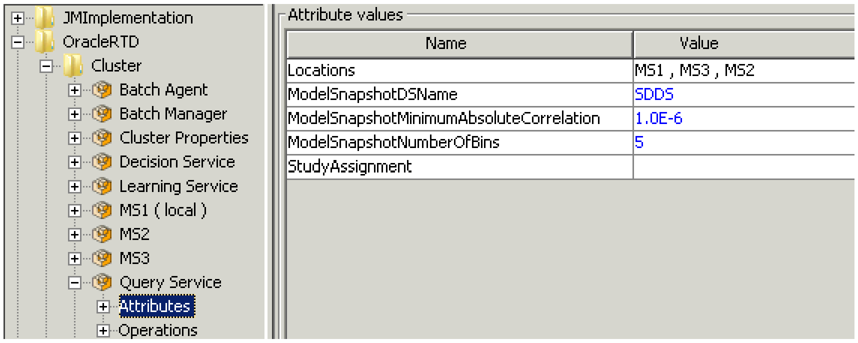
Saves all studies.

blockingModelBuild()

Process all outstanding learning records, then create and save prediction and learning models.

11.4.11 About OracleRTD > Cluster > Query Service

The Query Service MBean provides access to the following properties:



Name	Value
Locations	MS1 , MS3 , MS2
ModelSnapshotDSName	SDDS
ModelSnapshotMinimumAbsoluteCorrelation	1.0E-6
ModelSnapshotNumberOfBins	5
StudyAssignment	

The Cluster > Query Service MBean has the following attributes:

Attribute	Description
Locations	The locations of the running QS instances in the cluster.
ModelSnapshotDSName	The JNDI name of the data source for the model snapshots. Default: SDDS. A cluster-wide property, not overridden at the local level. If updated, the new value is used only after a restart of the Query Service.

Attribute	Description
ModelSnapshotMinimumAbsoluteCorrelation	Controls whether to dump all correlation rows or to set a minimum correlation value for dumping. A cluster-wide property, not overridden at the local level. The default value of 0.000001 prevents the dumping of very small value correlation rows. Set the value to 0 to dump all correlation rows. If updated, the new value is used on the next model snapshot.
ModelSnapshotNumberOfBins	Controls the number of bins for model snapshots. Numeric attribute values are automatically binned, or assigned to numeric ranges. The default number of bins is 5. To achieve greater resolution of your numeric data, increase the number of bins. A cluster-wide property, not overridden at the local level. If updated, the new value is used on the next model snapshot. Note that for the same numeric attribute, Oracle RTD creates different bins in different time windows. Therefore, it is unlikely that you will be able to join numeric attribute values across time windows.
StudyAssignment	Comma-separated list of QS instances and studies assigned to them *(format "Server1=Study1,Study2;Server2=Study3")*. The dummy study value "*" indicates that the QS instance can service all queries. "*" is the default for all QS instances. For more information, see Section 11.1.9.2, "Query Service Details."

The Cluster > Query Service MBean has the following operation:

incrementalSnapshot()

Saves a snapshot of this study to the database (delta).

completeSnapshot()

Saves a snapshot of this study to the database (complete).

deleteSnapshot()

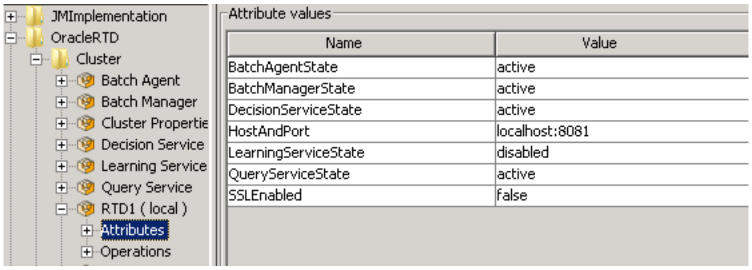
Removes this study's snapshot from the database.

restoreDefaultProperties()

Restores the system defaults for this MBean's properties.

11.4.12 About OracleRTD > Cluster > [Node]

Each Node in the cluster has a representative Node MBean, for example, the MBean for node RTD1:



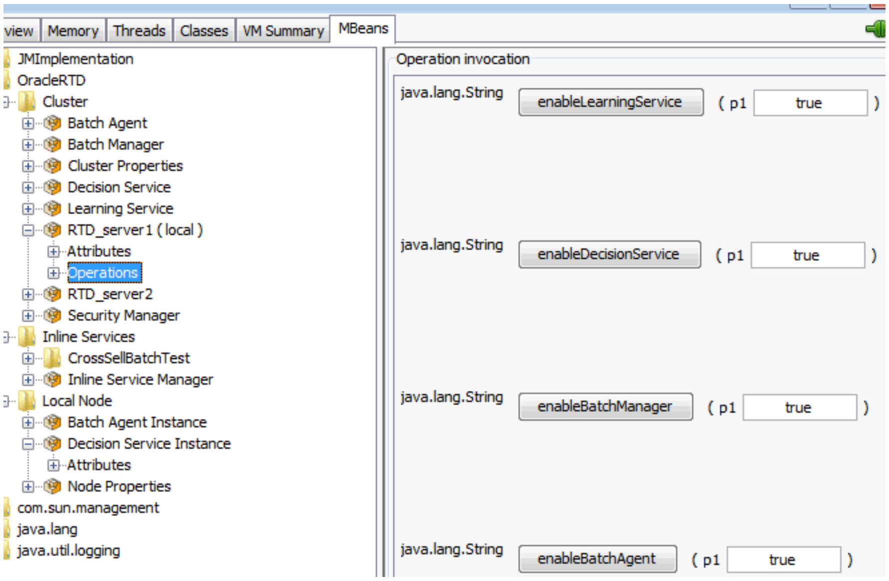
Name	Value
BatchAgentState	active
BatchManagerState	active
DecisionServiceState	active
HostAndPort	localhost:8081
LearningServiceState	disabled
QueryServiceState	active
SSLEnabled	false

The Service states for each of the Oracle RTD Services (Batch Agent, Batch Manager, Decision Service, Learning Service, Query Service) on the node are displayed.

The **HostAndPort** of the node is displayed. This is the node's value of the invariant and required JVM system property rtd.HTTPHostAndPort.

The **SSLEnabled** attribute indicates whether the port specified in HostAndPort is SSL enabled. This is the value of the invariant system property rtd.SSLEnabled, which defaults to false.

The Operations of the MBean permit enablement and disablement of each of the Service types on the node, as shown:



java.lang.String enableLearningService (p1 true)

java.lang.String enableDecisionService (p1 true)

java.lang.String enableBatchManager (p1 true)

java.lang.String enableBatchAgent (p1 true)

The Cluster > [Node] MBean has the following operations:

- **enableLearningService()**
- **enableDecisionService()**
- **enableBatchManager()**
- **enableBatchAgent()**
- **enableQueryService()**

Each of these operations has the **Enable** parameter.

This parameter can be set to **true** or **false** to enable or disable the appropriate Oracle RTD Service.

11.4.13 About OracleRTD > Cluster > Security Manager

The Cluster > Security Manager MBean has the following operations:

assignPermission()

Assigns the specified cluster permission. See Section 5.4, "Assigning Permissions" for more information. This operation accepts the following parameters:

Role: The role to which you want to assign permissions.

permCode: The code of the permission.

listDirectPermissions()

Returns the permissions directly assigned to a specific role. This operation accepts the following parameter:

Role: The target role.

listEveryoneHavingDirectPermissions()

Returns the names of the roles that have directly assigned permissions.

listPermissionCodes()

Returns the localized name and integer code for all possible permission types. The codes are used as operands for various operations that require a permission type designation.

removeAllDirectPermissions()

Removes all permissions directly assigned for the specified role. This operation accepts the following parameter:

role: The target role.

removePermission()

Removes the specified permission. This operation accepts the following parameters:

Role: The target role name.

permCode: The code of the permission you want to remove.

revertToStandardPermissions()

Reverts to the standard permissions.

11.4.14 About OracleRTD > Inline Services > InlineServiceManager

Each deployed Inline Service is displayed under the InlineServiceManager MBean.

The Inline Services > InlineServiceManager MBean has the following attribute:

Attribute	Description
InlineServices	List of deployed inline services.

The Inline Services > InlineServiceManager MBean has the following operations:

refreshInlineServiceMBeans()

Removes MBeans for applications no longer in the database, and creates MBeans for new ones.

removeAllInlineServices()

Removes all Inline Services (loaded, loadable, failed).

11.4.15 About OracleRTD > Inline Services > [Inline Service] > [Deployment State] > [Flag]

Inline Service MBeans can be viewed by choosing the name of an Inline Service, then choosing a Deployment State (for example, Development, QA, or Production), then choosing a particular flag (Failed, Inactive, or Loadable). For example: OracleRTD > Inline Services > CrossSell > Production > Loadable.

The MBeans for a specific Inline Service have the following attributes:

Attribute	Description
DeploymentState	Development, QA, or Production.
Flag	Failed, Inactive, or Loadable.
LockStatus	The lock status for the Inline Service.
ServiceId	The service ID for the Inline Service.

The MBeans for a specific Inline Service have the following operations:

unlockService()

Unlocks this service.

removeService()

Stops an Inline Service in this server and removes the service from the database.

flushStatistics()

Flushes all of the statistics for this service to the database.

deleteStatistics()

Flushes and deletes all of the statistics for this service from the database.

deleteChoiceHistory()

Deletes all of the choice history for this service from the database.

deleteAllOperationalData()

Deletes all of the operational data for this service from the database. This includes choice history, statistics, and the study.

deleteStudy()

Removes the study for this service.

11.4.16 Invoking Maintenance Operations

Maintenance Operations appear in the MBean tree under an Inline Service when both of the following conditions hold:

- The Inline Service includes one or more Maintenance Operations
- The Inline Service has the Deployment flag set to **Loadable**

Each Maintenance Operation appears in both of the **BroadcastAsyncOperations** and **DirectBlockingOperations** nodes, under the **Loadable** entry.

Operations listed in the **DirectBlockingOperations** node are invoked on the local server only, and they return only after the operation has completed. The returned value will be displayed to the JConsole user in a popup dialog. If the operation has return type "void," then "null" will appear. If the operation fails for any reason, a short error message will be displayed in a JConsole popup dialog, and a more detailed report can be found in the log of that server.

Operations listed in the **BroadcastAsyncOperations** node are invoked across every node of a cluster. The operation returns immediately with the number of cluster members who received the broadcast. If the cluster has just one node, the operation returns 1, and the invocation is still asynchronous.

The following run-time considerations apply for Maintenance Operations:

- Oracle RTD does not guarantee that all cluster members are notified of Maintenance Operation invocations, although usually they will be.

  For example, if one member of a cluster is down when a Maintenance Operation is invoked, there is no notification to indicate that the cluster member should run the Maintenance Operation when it comes back up.
- Ordering of Maintenance Operations is not guaranteed. For example, if two Maintenance Operations A and B are invoked in sequence, an Inline Service may run B before it runs A, or it may even run them simultaneously.

11.4.17 About OracleRTD > Learning Server > [Study]

The Learning Server > [Study] MBeans have the following attributes:

Attribute	Description
Models	Models belonging to this study.
Name	The name of the study.

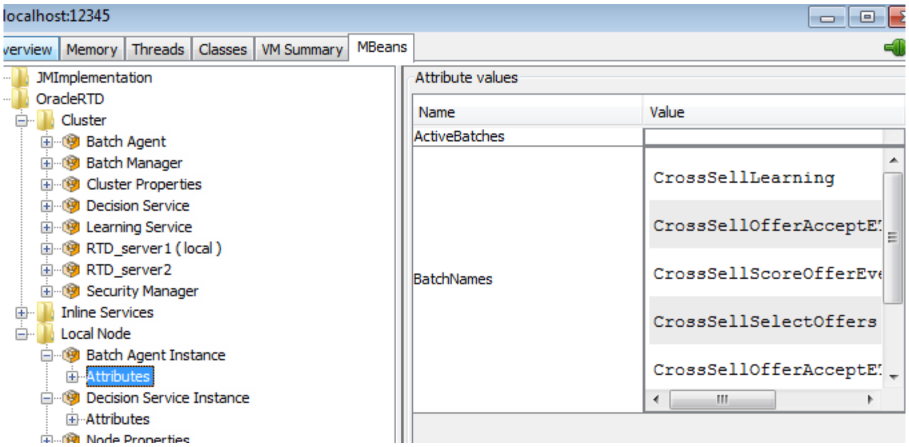
The Learning Server > [Study] MBeans have the following operation:

Delete()

Deletes this study.

11.4.18 Local Node - Batch Agent Instance MBean

The Local Node > Batch Agent Instance MBean displays the running batch jobs and names on the BA instance:



Name	Value
ActiveBatches	
BatchNames	CrossSellLearning CrossSellOfferAcceptE CrossSellScoreOfferEve CrossSellSelectOffers CrossSellOfferAcceptE

The Local Node > Batch Agent Instance MBean has the following attributes:

Attribute	Description
ActiveBatches	List of all batch jobs currently running on this batch agent, paused, or waiting to run. The list could be empty.
BatchNames	List of batches registered with this batch agent.

11.4.19 About OracleRTD > Local Node > Decision Service Instance

The Local Node > Decision Service Instance MBean has the following locally overrideable attributes:

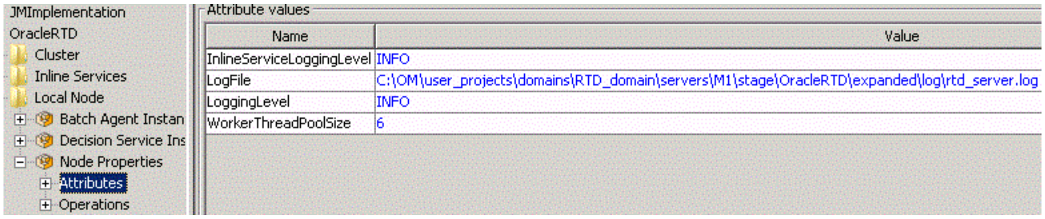
Attribute	Description
CurrentReceivedRequestsForwarded	Number of requests that were forwarded from other servers to this server, and which have not yet been completely processed by this server.
CurrentRequestsForwarded	Number of requests that have been forwarded from this server to other servers, and for which no acknowledgment has yet been received to indicate that the request has been processed by the forwarded-to server.
CurrentRequestsQueued	Number of currently waiting requests, not yet running.
CurrentRequestsRunning	Number of Integration Point requests that are currently being processed by Inline Services.
CurrentSessions	Number of Decision Service sessions still open.
[The general notes (see Description) apply to the following DSPerf... attributes for this MBean.]	General notes about the DSPerf... attributes ▪ All are node-scoped properties only, and can be set in the Decision Service Instance MBean or using a JVM system property. ▪ All Oracle RTD JVM system properties take the prefix "rtd.", for example rtd.DSPerfCounterAppend. ▪ All are dynamic reconfiguration properties - restart of the node is not required.

Attribute	**Description**
DSPerfCounterAppend	Whether to append to the DS performance CSV file or not. Default is TRUE. (See also *General notes about the DSPerf... attributes* in previous Description.)
DSPerfCounterEnabled	Whether to enable DS performance logging to a CSV file or not. Default is FALSE. (See also *General notes about the DSPerf... attributes* in previous Description.)
DSPerfCounterFileRollingEnabled	If set to TRUE, then the filename in DSPerfCounterLogFile is expected to include a date format string, such as RTD1_ds_perf_%yyyy-MM-dd_hh.csv. At runtime, the date format string will be replaced with current date and time values. The filename will thus change as the hour changes and a new performance log file begins. The filename MUST include a % if DSPerfCounterFileRollingEnabled is true, indicating the start of the date format string. The date format string can be anything allowed by the Java class SimpleDateFormat (http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html). Note that the date format should not be too granular - for example, if it includes a minute element, the log file will roll every minute.
DSPerfCounterLogFile	File name to write DS performance data into. Default is `RTD_HOME/log/ds_perf.csv`. (See also *General notes about the DSPerf... attributes* in previous Description.)
DSPerfCounterLogFileInterval	The interval at which the DS performance log is written, in mSec. Default is 5000. (See also *General notes about the DSPerf... attributes* in previous Description.)
IntegrationPointMaxConcurrentJobs	The maximum number of concurrently executing Integration Point requests. This should normally be set to `0`, in which case the value is calculated as follows, where `Math.ceil` means "round up to the next higher integer:" `NumCPUs * Math.ceil(1/(1-DSRequestIOFactor)) + 5`
IntegrationPointQueueSize	The maximum number of Integration Point requests that can wait to execute. If a request tries to exceed this number, the server terminates the request with a "Server Too Busy" error message. This setting should be less than or equal to the number of servlet threads allocated by the servlet container minus the configured or calculated value of IntegrationPointMaxConcurrentJobs.
MaxAllowedConcurrentRequests	Maximum number of requests that can run concurrently.
NumCPUs	Number of CPUs in this host.
PeakReceivedRequestsForwarded	Largest number of received requests forwarded.
PeakRequestsForwarded	Largest number of requests forwarded.

Attribute	Description
PeakRequestsQueued	Largest number of requests that have had to wait at any one time.
ReceivedRequestsForwarded	Total number of requests that were forwarded from other servers to this server.
RequestQueueCapacity	Maximum number of requests that can wait concurrently.
RequestsForwarded	Total number of requests forwarded to another server in the cluster.
RequestsQueued	Total number of requests that have had to wait before running.
RequestsWhenQueueFull	Total number of requests rejected because the request queue was full.
TimedOutRequests	Total number of requests that have timed out.
TotalRequests	Total number of requests seen since the server started.
TotalSessions	Total number of Decision Service sessions created.

11.4.20 About OracleRTD > Local Node > Node Properties

The Local Node > Node Properties MBean has the following node-scoped properties:



Name	Value
InlineServiceLoggingLevel	INFO
LogFile	C:\OM\user_projects\domains\RTD_domain\servers\M1\stage\OracleRTD\expanded\log\rtd_server.log
LoggingLevel	INFO
WorkerThreadPoolSize	6

The **LoggingLevel**, **InlineServiceLoggingLevel** and **WorkerThreadPoolSize** properties override, for the local node, those defined in the Cluster Properties MBean for the local node.

LogFile - The log file to use.

Default log file name is:

- <*RTD_HOME*>/log/<*instance_name*>/rtd_server.log

This is a node-scoped property only, which may be set in the Node Properties MBean or using a JVM system property.

Updates do not require a node restart.

The previous name for this property was SDLoggingFileName.

The Operations include a **restoreDefaultProperties()** operation to restore the system defaults for these properties.

11.4.21 About OracleRTD > Study > [Study Name] > [Model]

The Study > [Study Name] > [Model] MBeans have the following attributes:

Attribute	Description
Attributes	Names of the model attributes. The names listed here match the attribute names in the session for your Inline Service.
Name	The name of the model.
TimeWindows	List of the ranges of time that have been learned about by this particular model.

The Study > [Study Name] > [Model] MBeans have the following operations:

Delete()

Deletes this model.

DeleteAttributeValue()

Erases model data collected for a value of an attribute. This operation accepts the following parameters:

AttributeName: The name of an attribute.

Value: The value to be deleted.

DeleteAttributeValueRange()

Erases model data collected for a range of values of an attribute. This operation accepts the following parameters:

AttributeName: The name of an attribute.

HighestValue: The highest value to be deleted.

LowestValue: The lowest value to be deleted.

StartNewTimeWindow()

Closes the current time window and starts a new one. Do not use this operation in a production environment, because it may impair future model learning.

11.5 Required System Properties and Other Requirements

This section describes mandatory and optional requirements for Oracle RTD systems. Some of these apply to all supported application servers, others are application-server specific.

There are two system properties that are required for all Oracle RTD systems:

- **rtd.instanceName** - Must be non-null and unique over the cluster. Only the following characters are allowed: alphanumerics, underscore, dash, period. Any others and the cluster will not start.
- **rtd.HTTPHostAndPort** - Must be in *<host>:<port>* format - non-null and unique over the cluster.

Also required for all systems: the EMF registry must be set:

- org.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.SDEMFRegistry

The following are optional or application-server specific requirements:

- On Sun JVM on WebLogic systems only, the PermGen size must be at least 256m: -XX:MaxPermSize=256m
- Required for WebLogic systems only: weblogic.wsee.skip.async.response=true
- Optional for all systems: sun.reflect.inflationThreshold=2147483647

  This only applies to Decision Servers. If an Inline Service uses many external rules, specifying this parameter may improve their initial load time.

A

System Log and Configuration Files

This appendix provides the names and locations of the Oracle RTD log and configuration files. To ensure the integrity of the information in these files, you should use operating system or other tools to set the appropriate file permissions on the parent directories that contain these files, so that an intruder cannot access them.

This section contains the following topics:

- Section A.1, "Encoding for Localized Characters"
- Section A.2, "Log Files"
- Section A.3, "Configuration Files"

A.1 Encoding for Localized Characters

In order for localized characters (such as inline service choice or attribute names) to display properly in the log files, these files should have UTF-8 encoding. Add the JVM environment variable `file.encoding=utf8` (for example, `-Dfile.encoding=utf8`) to the JVM configuration either by adding the statement to the server startup script or by invoking the parameters at JVM startup.

A.2 Log Files

Oracle RTD provides both server and client logs. In addition, there are log files specific to Eclipse and application servers.

This section contains the following topics:

- Section A.2.1, "Main Oracle RTD Log Files"
- Section A.2.2, "Log Files for Oracle RTD Client Tools"
- Section A.2.3, "Server-Side Application Server Specific Log Files"
- Section A.2.4, "Eclipse Log File"

A.2.1 Main Oracle RTD Log Files

Table A–1 shows the two main Oracle RTD log files. The Oracle RTD Server log is the main log to use for troubleshooting problems.

Table A–1 Main Oracle RTD Logs

Log Type	Default Location
Oracle RTD Server log	`*RTD_RUNTIME_HOME*/log/*INSTANCE_NAME*_rtd_server.log`

Table A–1 (Cont.) Main Oracle RTD Logs

Log Type	Default Location
Oracle RTD Client log	*RTD_HOME*/log/client.log

where:

- *RTD_HOME* is the directory into which you extract the Oracle RTD files during installation.
- *RTD_RUNTIME_HOME* is the directory into which you install runtime Oracle RTD.

The directory *RTD_RUNTIME_HOME* is application server dependent, as shown in Table A–2.

Table A–2 Application Server Specific RTD_RUNTIME_HOME Directories

Application Server	*RTD_RUNTIME_HOME* Directory
WebSphere	*WEBSPHERE_HOME*/AppServer/profiles/*profile_name*/installedApps/*host_node_cell_name*/OracleRTD.ear/
WebLogic	*MW_HOME*/user_projects/domains/*domain_name*/servers/*server_name*/stage/OracleRTD/

Setting Logging Levels for the Oracle RTD Server Log

To change logging options for the Oracle RTD Server log, use JConsole. See Section 11.4.8, "About OracleRTD > Cluster > Cluster Properties" and Section 11.4.20, "About OracleRTD > Local Node > Node Properties" for more information.

Error Messages and Exceptions in Oracle RTD Log Files

For exception messages that appear in Oracle RTD log files, see Appendix B, "Exceptions".

For error messages that may occur during Inline Service development, see Appendix A, "Development Error Messages" in *Oracle Real-Time Decisions Platform Developer's Guide*.

A.2.2 Log Files for Oracle RTD Client Tools

In addition to the Oracle RTD Client log, Oracle RTD maintains the following client tool log files:

- *RTD_HOME*/scripts/SDDBTool.log
- *RTD_HOME*/log/loadgen.csv

A.2.3 Server-Side Application Server Specific Log Files

This section lists the server-side log files specific to the application servers supported by Oracle RTD.

A.2.3.1 Server-Side Log Files for WebSphere

```
WEBSPHERE_HOME/AppServer/profiles/profile_name/logs/server1/startServer.log
WEBSPHERE_HOME/AppServer/profiles/profile_name/logs/server1/stopServer.log
WEBSPHERE_HOME/AppServer/profiles/profile_name/logs/server1/SystemOut.log
WEBSPHERE_HOME/AppServer/profiles/profile_name/logs/server1/SystemErr.log
WEBSPHERE_HOME/AppServer/profiles/profile_name/logs/fdcc/server1/
```

```
server1_*.txt (error logs)
```

A.2.3.2 Server-Side Log Files for WebLogic

```
MW_HOME/user_projects/domains/domain_name/servers/server_name/logs/
server_name.log
MW_HOME/user_projects/domains/domain_name/servers/server_name/logs/
server_name.out
```

The file server.log contains all the Oracle RTD log information as well as additional server log information.

A.2.4 Eclipse Log File

Log information specific to Eclipse can be found in the Eclipse log, located at:

- `Studio_Workspace\.metadata\.log`

where *`Studio_Workspace`* is user-configurable from Version 3.0 onwards. In general, the default location for *`Studio_Workspace`* is `C:\Users\ <windows_user_name>\Oracle RTD Studio\.`

For example, `C:\Users\jsmith\Oracle RTD Studio\.metadata\.log`.

Setting Logging Levels for Eclipse

To set the logging levels for Eclipse, edit the following file:

```
RTD_HOME\eclipse\plugins\com.sigmadynamics.studio_
3.2.0.0.0\etc\eclipse-log.properties
```

To adjust logging levels, set the values to `true` or `false`. The default settings are as follows:

- debug=false
- info=true
- warn=true
- error=true
- fatal=true
- trace=false

A.3 Configuration Files

This section lists the configuration files specific to the application servers supported by Oracle RTD.

Oracle RTD Configuration Files for WebSphere

```
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_cell_name/
OracleRTD.ear/etc/sdconfig.xml
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_cell_name/
OracleRTD.ear/etc/sdlog4j.xml
```

Oracle RTD Configuration Files for WebLogic

```
MW_HOME/user_projects/domains/domain_name/servers/server_
name/stage/OracleRTD/etc/sdconfig.xml
MW_HOME/user_projects/domains/domain_name/servers/server_
name/stage/OracleRTD/etc/sdlog4j.xml
```

B

Exceptions

This appendix shows the exceptions that may occur during the running of Oracle RTD processes.

Table B–1 lists the exceptions that may appear as messages in your Oracle RTD log files. Suggested actions are shown for each exception. If these actions do not resolve your problem, contact Oracle Support Services.

> **Note:** For error messages that may occur during Inline Service development, see Appendix A, "Development Error Messages" in *Oracle Real-Time Decisions Platform Developer's Guide*.

Table B–1 List of Exceptions

Exception	Action
Error reading file *file_name*	Check that the file exists and has read permissions.
Error writing file *file_name*	Check that the file exists and has write permissions.
Cannot load Inline Service with id *id*. No Decision Service is present.	Check that the server where the Decision Service is deployed is running. Use JConsole to find where the Decision Service is deployed. For information about JConsole, see Chapter 11, "Managing Oracle Real-Time Decisions".
Failure setting up Smart Client's default responses from file, *file_name*	Check the Smart Client properties file for the location of the file and check to make sure it exists.
Error connecting to server	Check your server connectivity and that your network is working properly.
There were compiler errors: *errors*	Compiler errors may occur on the server if you have configured an Inline Service in a newer version of Decision Studio than the version of the Server. Upgrade your Server version to correct the problem.
Internal Error.	Contact Oracle Support Services.
No default choice is defined for Inline Service *Inline_Service_name*, or its Integration Point *Integration_Point_name*.	Default choices are choices used by the calling application when the server is unavailable. Default choices are defined at the Integration Point level.
Could not create a backup of {0} in {1}	If the backup location is on your network, check the network connectivity. If the backup location is local, check that you have enough disk space for the backup.

Table B–1 (Cont.) List of Exceptions

Exception	Action
Could not find JMX Server *server_name*	If the JMX Server is unavailable, check the database connectivity and make sure the server is running.
Unknown Decision Service message type received by HTTP endpoint: *endpoint_name*	Refer to Part II, Integration with Oracle RTD, in *Oracle RTD Platform Developer's Guide* for examples and specifications for HTTP queries.
Malformed SOAP query	Refer to Part II, Integration with Oracle RTD, in *Oracle RTD Platform Developer's Guide* for examples and specifications for SOAP queries.
Session merging not implemented, but found two keys in same request referencing different sessions	Session merging is not yet implemented. In your Inline Service, use only one session key.
Input column at location *table_or_stored_procedure* and name *column_name* has a null value which is not supported in where clauses	The Input column of a where clause cannot be null. Go to the data source indicated and provide a value for the column.
The current server has a newer version of the metadata so *Inline_Service* cannot be loaded	This error can occur if you are using a newer version of Decision Studio with an older version of Real-Time Decision Server. Upgrade your server to correct the error.
The current server *server_name* supports metadata versions up to *version_number*, but the metadata of *Inline_Server* is at version *version_number*	This error can occur if you are using a newer version of Decision Studio with an older version of Real-Time Decision Server. Upgrade your server to correct the error.
Generation of Inline Service "*Inline_Service_name*" failed	This error can occur if you are using a newer version of Decision Studio with an older version of Real-Time Decision Server. Upgrade your server to correct the error.
Unable to read a study definition created by a newer version of the software.	This error can occur if you are using a newer version of Decision Studio with an older version of Real-Time Decision Server. Upgrade your server to correct the error.
Unable to read a prediction model created by a newer version of the software.	This error can occur if you are using a newer version of Decision Studio with an older version of Real-Time Decision Server. Upgrade your server to correct the error.
Encountered a database record created by a newer version of the software.	This error can occur if you are using a newer version of Decision Studio with an older version of Real-Time Decision Server. Upgrade your server to correct the error.
Unable to read a learning model created by a newer version of the software.	This error can occur if you are using a newer version of Decision Studio with an older version of Real-Time Decision Server. Upgrade your server to correct the error.
No result set found while getting result set of procedure "*stored_procedure_name*".	This error can occur if you have defined a result set for your data source, but there is none on the stored procedure. Check the stored procedure definition in the database.
Generic Exception caught while setting blob for procedure "*stored_procedure_name*".	This error can occur for several reasons. Check your database connectivity. Check that the database server is running.

Table B–1 (Cont.) List of Exceptions

Exception	Action
Exception during output of batched statements: database *insert/update/select/delete* operation for *table_or_stored_procedure* took duration. Batch size is *number_of_results*.	This error can occur for several reasons. Check your database connectivity. Check that the database server is running.
The stored procedure "*stored_procedure_name*" was not found in database.	This error can occur for several reasons. Check your database connectivity. Check that the database server is running. Finally, check that the stored procedure named in the data source is named correctly and resident on the database.
Failed to find column "*column_name*" in table "*table_name*".	This error can occur for several reasons. Check your database connectivity. Check that the database server is running. Finally, check that the table named in the data source is named correctly and resident on the database.
Error setting up Smart Client properties.	This error can occur if you have not properly configured your Smart Client properties file. Refer to Part II, Integration with Oracle RTD, in *Oracle RTD Platform Developer's Guide* for information on using the Smart Client.
Failed to load Inline Service: *Inline_Service_name*	This error can occur on start-up of Real-Time Decision Server. Redeploy the Inline Services that did not start up. If the error reoccurs, contact Oracle Support Services.