Oracle® Real-Time Decisions

Installation and Administration Guide

Version 3.0.0.1

E13856-02

April 2011

Oracle Real-Time Decisions Installation and Administration Guide, Version 3.0.0.1

E13856-02

Contents

5 Configuring WebLogic for Oracle Real-Time Decisions

6 Configuring JBoss for Oracle Real-Time Decisions

7 Configuring Security for Oracle Real-Time Decisions

8 Configuring Data Access for Oracle Real-Time Decisions

9 Additional Configuration Settings and Starting Client Tools

10 Production Deployment of Oracle Real-Time Decisions

11 Command Line Deployment of Inline Services

12 Setting Up and Using Model Snapshots

13 Deploying Oracle RTD into an Oracle RTD Cluster

14 Performance Monitoring

15 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 Version 3.0.0.1. 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

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

To reach AT&T Customer Assistants, dial 711 or 1.800.855.2880. An AT&T Customer Assistant will relay information between the customer and Oracle Support Services at 1.800.223.1711. Complete instructions for using the AT&T relay services are available at `http://www.consumer.att.com/relay/tty/standard2.html`. After the AT&T Customer Assistant contacts Oracle Support Services, an Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process.

Related Documents

For more information, see the following documents in the Oracle RTD Version 3.0 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.

1 Preparing for Installation

Oracle RTD is supported on both Windows and UNIX platforms, and runs on Oracle Containers for J2EE (OC4J), Oracle WebLogic, IBM WebSphere, and JBoss 4.3 EAP application servers. For detailed information about supported platforms, refer to *Oracle Real-Time Decisions System Requirements and Supported Platforms* for Versions 3.0 and 3.0.0.1.

> **Note:** If you plan to install Oracle RTD on OC4J, you can choose either a standalone version of OC4J, or the version of OC4J that comes with Oracle Application Server.

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 Java Development Kit (JDK)"
- Section 1.4, "Installing the Application Server"
- 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, Java Development Kit, and application server.

2. Install files for the Real-Time Decision Server and Oracle RTD clients, and create and initialize the database.
3. Configure the data source for the Oracle RTD Database.
4. Deploy and configure Real-Time Decision Server using the administration console for your application server, and other tools.
5. Configure authentication settings.
6. 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, Microsoft SQL Server, or IBM DB2. 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 Versions 3.0 and 3.0.0.1, 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 Java Development Kit (JDK)

For some configurations, you need to download and install a Java Development Kit (JDK) before proceeding with Oracle RTD installation. The following sections provide more information about whether you need to install a JDK, and, if so, which JDK to install.

After installing the 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;'
```

1.3.1 Standalone OC4J Installations

If you plan to install Real-Time Decision Server on standalone OC4J on **Windows**, you must install Sun JDK version 1.5.0.8+ or 1.6 before proceeding with the installation of Oracle RTD.

If you plan to install Real-Time Decision Server on standalone OC4J on **Solaris** or **Linux**, you must install Sun JDK version 1.5.0.8+ or 1.6 before proceeding with the installation of Oracle RTD.

If you plan to install Real-Time Decision Server on standalone OC4J on **HP-UX**, you must install HP-UX JDK 5.0.05 before proceeding with the installation of Oracle RTD.

If you plan to install Real-Time Decision Server on standalone OC4J on **AIX-based systems**, you must install IBM Java 5 SR2 before proceeding with the installation of Oracle RTD.

You must install Sun JDK version 1.5.0.8+ or 1.6 on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool). You can also use the JDK shipped within the Oracle RTD archive file, in the `jdk` folder under the Oracle RTD home directory, for example, `C:\OracleBI\RTD\jdk`.

1.3.2 Oracle Application Server Installations

If you plan to install Real-Time Decision Server on the version of OC4J that ships with Oracle Application Server on **Windows**, you do not need to download a separate JDK. Oracle Application Server comes with its own JDK. However, you do need to set the `JAVA_HOME` system environment variable and set its value to the full path name of the install location of the JDK being used by Oracle Application Server.

If you plan to install Real-Time Decision Server on the version of OC4J that ships with Oracle Application Server on **Solaris**, **Linux**, **HP-UX**, or **AIX-based systems**, you do not need to download a separate JDK. Oracle Application Server comes with its own JDK.

You must install Sun JDK version 1.5.0.8+ or 1.6 on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool). You can also use the JDK shipped within the Oracle RTD archive file, in the `jdk` folder under the Oracle RTD home directory, for example, `C:\OracleBI\RTD\jdk`.

1.3.3 WebSphere Installations

If you plan to install Real-Time Decision Server on WebSphere on **Windows**, you do not need to download a separate JDK. WebSphere comes with its own JDK. However, you do need to set the `JAVA_HOME` system environment variable and set its value to the full path name of the install location of the JDK being used by WebSphere.

If you plan to install Real-Time Decision Server on WebSphere on **Solaris**, **Linux**, **HP-UX**, or **AIX-based systems**, you do not need to download a separate JDK. WebSphere comes with its own JDK.

You must install Sun JDK version 1.5.0.8+ or 1.6 on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool). You can also use the JDK shipped within the Oracle RTD archive file, in the `jdk` folder under the Oracle RTD home directory, for example, `C:\OracleBI\RTD\jdk`.

1.3.4 WebLogic Installations

If you plan to install Real-Time Decision Server on WebLogic on **Windows**, **Solaris**, or **Linux**, you do not need to download a separate JDK. WebLogic comes with its own JDK.

You must install Sun JDK version 1.5.0.8+ or 1.6 on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool). You can also use the JDK shipped within the Oracle RTD archive file, in the jdk folder under the Oracle RTD home directory, for example, C:\OracleBI\RTD\jdk.

> **Note:** If you are installing WebLogic on a UNIX system, you must add the bin directory of the appropriate JDK to the beginning of the PATH operating system environment variable, as follows:
>
> ```
> PATH=JAVA_HOME/bin:$PATH
> export PATH
> ```

1.3.5 JBoss Installations

If you plan to install Real-Time Decision Server on JBoss, download and install JDK version 1.6, then set the *JAVA_HOME* system environment variable to the folder where JDK was installed.

If you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool on a separate machine from the Real-Time Decision Server, Oracle recommends that you install JDK version 1.5 on that separate machine. After installing the JDK, set the *JAVA_HOME* system environment variable to the folder where JDK was installed. You can also use the JDK shipped within the Oracle RTD archive file, in the jdk folder under the Oracle RTD home directory, for example, C:\OracleBI\RTD\jdk.

1.4 Installing the Application Server

Real-Time Decision Server runs on one of the following application servers: OC4J (a component of Oracle Application Server), WebSphere, WebLogic, or JBoss 4.3 EAP. For detailed information about supported application server versions, refer to *Oracle Real-Time Decisions System Requirements and Supported Platforms* for Versions 3.0 and 3.0.0.1, available from the Oracle RTD Documentation Web site at:

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

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

1.4.1 Installing OC4J for Oracle RTD

You can choose to install a standalone version of OC4J, or the version of OC4J that comes with Oracle Application Server. The standalone version of OC4J includes a single server in a small package without an installer. The version of OC4J included with Oracle Application Server provides a full-suite installer and supports J2EE clustering.

1.4.1.1 Installing Standalone OC4J

Follow these steps to install standalone OC4J for Oracle RTD:

1. On the computer where you want to run Real-Time Decision Server, install standalone OC4J. Refer to *Oracle Containers for J2EE Configuration and Administration Guide* for information about installing standalone OC4J.

> **Note:** In this document, the directory where the standalone OC4J instance is installed is referred to as *OC4J_HOME*. For example, if you unzip the standalone OC4J files into `C:\Program Files`, your *OC4J_HOME* will be `C:\Program Files\OC4J`.

2. Make a backup copy of the file *OC4J_HOME*`\bin\oc4j.cmd` (or *OC4J_HOME*`/bin/oc4j` on UNIX).
3. Open OC4J_HOME`\bin\oc4j.cmd` (or OC4J_HOME`/bin/oc4j` on UNIX) for editing. At the top of the file, add the following line:

   ```
   set ORACLE_HOME=your_OC4J_HOME
   ```

   For example:

   ```
   set ORACLE_HOME=C:\OracleBI\OC4J
   ```

   Then, save and close the file.
4. Start OC4J using the `oc4j` script. To do this, open a command prompt and go to *OC4J_HOME*`\bin` (or *OC4J_HOME*`/bin` on UNIX), then type `oc4j -start`.
5. When prompted, enter a new password and confirm it. This password is the password for the `oc4jadmin` account, which is the administrative account for the OC4J instance. You only need to provide this password the first time you start OC4J; you do not need to enter it for subsequent OC4J startups.
6. Shut down the OC4J server using the `oc4j` script. To do this, open a command prompt and go to *OC4J_HOME*`\bin` (or *OC4J_HOME*`/bin` on UNIX), then issue the following command. You must supply the ORMI port used by OC4J, which is `23791` by default, as well as the password for the `oc4jadmin` account.

   ```
   oc4j -shutdown -port oc4jOrmiPort -password oc4jadmin_password
   ```

   For example:

   ```
   oc4j -shutdown -port 23791 -password mypwd
   ```

1.4.1.2 Installing Oracle Application Server

On the computer where you want to run Real-Time Decision Server, install Oracle Application Server. Refer to *Oracle Application Server Installation Guide* for information about installing Oracle Application Server.

> **Note:** In this document, the directory where Oracle Application Server is installed is referred to as *ORACLE_AS_HOME*. For example, if you install Oracle Application Server into `C:\Oracle`, your *ORACLE_AS_HOME* will be similar to `C:\Oracle\10.1.3.1\OracleAS_1`.

1.4.2 Installing WebSphere for Oracle RTD

Follow these steps to install WebSphere for Oracle RTD:

1. Install WebSphere on the computer where you want to run Real-Time Decision Server. Refer to the WebSphere documentation for information about installing WebSphere. When installing, you must be logged into the local computer as an administrative user.

Note: In this document, the directory where WebSphere is installed is referred to as *WEBSPHERE_HOME*. For example, on Windows, the default *WEBSPHERE_HOME* is `C:\Program Files\IBM\WebSphere`.

1.4.3 Installing WebLogic for Oracle RTD

Follow these steps to install WebLogic for Oracle RTD:

1. If you are installing WebLogic on UNIX, make sure that the *JAVA_HOME*`/bin` directory for the appropriate JDK has been added to the beginning of the `PATH` operating system environment variable. See Section 1.3.4, "WebLogic Installations" for more information about JDK requirements for WebLogic.
2. Install WebLogic on the computer where you want to run Real-Time Decision Server. Refer to the WebLogic documentation for information about installing WebLogic.

   When the Installation Complete window appears, you can select Run Quickstart, and click Done to start the WebLogic Server.

Note: In this document, the directory where WebLogic is installed is referred to as *MW_HOME*. For example, if you install WebLogic into `C:\`, your *MW_HOME* will be similar to `C:\Oracle\Middleware`.

1.4.4 Installing JBoss for Oracle RTD

Follow these steps to install JBoss 4.3 EAP for Oracle RTD:

1. Install JBoss 4.3 EAP on the computer where you want to run Real-Time Decision Server.

Note: The directory into which you installed JBoss 4.3 EAP is referred to as *JBOSS_HOME*.

For example, `C:\Program Files\EnterprisePlatform-4.3.0.GA_CP03, C:\jboss-eap-4.3, d:\genesis\jboss-eap-4.3`.

2. Enable the JBoss default user, as follows:
   a. Open the file *JBOSS_HOME* `/jboss-as/server/default/conf/props/jmx-console-users.properties`
   b. Uncomment the sample user/password (if needed): `#admin=admin`

      Example: `admin=admin`
   c. Save the file.
3. Set the JBoss HTTP and HTTPS ports, as follows:
   a. Open the file *JBOSS_HOME* `/jboss-as/server/default/deploy/jboss-web.deployer/server.xml`
   b. For HTTP, near the top of the file at "`<Connector port="8080" ...`'

add the following line after the line "`enableLookups`":

`URIEncoding="UTF-8"`

For example:

```
<Connector port="8080" address="${jboss.bind.address}"
          maxThreads="250" maxHttpHeaderSize="8192"
          emptySessionPath="true" protocol="HTTP/1.1"
          enableLookups="false" redirectPort="8443" acceptCount="100"
          URIEncoding="UTF-8"
          connectionTimeout="20000" disableUploadTimeout="true" />
```

c. For HTTPS, near the top of the file, replace:

```
<!--
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
                  maxThreads="150" scheme="https" secure="true"
                  clientAuth="false" sslProtocol="TLS" />
-->
```

with:

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
                  maxThreads="150" scheme="https" secure="true"
                  clientAuth="false" sslProtocol="TLS"
                  minSpareThreads="5" maxSpareThreads="15"
                  address="${jboss.bind.address}"
keystoreFile="${rtd.install.dir}/etc/ssl/sdserver.keystore"
         URIEncoding="UTF-8"
                  keystorePass="tc-ssl" />
```

d. Save the file.

4. Set the JBoss Web Container to use Java compiler compliance level 1.5, as follows:

   a. Open the file *`JBOSS_HOME`* `/jboss-as/server/default/deploy/jboss-web.deployer/conf/web.xml`

   b. Locate:

```
<servlet>
          <servlet-name>jsp</servlet-name>
```

   c. Add the following servlet `init` parameters:

```
<init-param>
   <param-name>compilerSourceVM</param-name>
   <param-value>1.5</param-value>
</init-param>
<init-param>
   <param-name>compilerTargetVM</param-name>
   <param-value>1.5</param-value>
 </init-param>
```

   d. Save the file.

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

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, "Quick Zip Installation"
- Section 2.6, "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.

   On the Oracle RTD installation DVD, the Oracle RTD archive files have different names, depending on operating system platform and application server type. For example, the Oracle RTD archive file for WebLogic on Windows is named `rtd_3.0.0_WL_win.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.

   On the Oracle RTD installation DVD, the Oracle RTD archive files have different names, depending on operating system platform and application server type. For example, the Oracle RTD archive file for WebSphere on UNIX is named `rtd_3.0.0_WAS_unix.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.

   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.

   On the Oracle RTD installation DVD, the Oracle RTD archive files have different names, depending on operating system platform and application server type. For example, the Oracle RTD archive file for OC4J (and Oracle Application Server) on Windows is named `rtd_3.0.0_OC4J_win.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* 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 standalone OC4J, *RTD_RUNTIME_HOME* is *OC4J_HOME*/j2ee/home/applications/OracleRTD.
- If you are using Oracle Application Server, *RTD_RUNTIME_HOME* is *ORACLE_AS_HOME*/j2ee/*oc4j_instance*/applications/OracleRTD.

> **Note:** The value *oc4j_instance* refers to the particular instance of OC4J where Oracle RTD has been deployed. For example, if Oracle RTD was deployed to the default OC4J instance, the path is: *ORACLE_AS_HOME*/j2ee/home/applications/OracleRTD.

- 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/.
- If you are using JBoss, *RTD_RUNTIME_HOME* is *JBOSS_HOME*/jboss-as/server/default/tmp/deploy/tmp<...>RTD.ear-contents, where <...> is a random number. Note that the runtime files server.log and log ds_perf.cvs are in RTD_HOME.

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 a Database: Microsoft SQL Server"
- Section 2.2.2, "Creating a Database: Oracle Database"
- Section 2.2.3, "Creating a Database: IBM DB2"
- Section 2.2.4, "Initializing the Oracle RTD Database Using SDDBTool"
- Section 2.2.5, "Initializing the Oracle RTD Database Using Command Line Scripts"

2.2.1 Creating a Database: Microsoft SQL Server

After installing SQL Server, use the SQL Server Enterprise Manager to create a new database.

To use SQL Server as your Oracle RTD Database, you must assign two database users to this database: an administration user and a run-time user. The administration user must have the server role `System Administrator`. The run-time user must have the database roles `db_ddladmin`, `db_reader`, and `db_writer`. Refer to the Microsoft SQL Server documentation for information about creating the database and assigning users.

You must install SQL Server Client Tools on the computer where you plan to run SDDBTool. See Section 2.2.4, "Initializing the Oracle RTD Database Using SDDBTool" for more information about SDDBTool.

2.2.2 Creating a Database: Oracle Database

To use Oracle Database as 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.

For Oracle Database, 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 2.2.4, "Initializing the Oracle RTD Database Using SDDBTool" for more information about SDDBTool.

2.2.3 Creating a Database: IBM DB2

After installing DB2 Server, use DB2 Control Center to create a new database. The code set of this database must be set to UTF-8.

To use DB2 as your Oracle RTD Database, you must provide two database users: an administration user and a run-time user. The administration user must have the `Database Administrator` authority on the database you created. The run-time user must have the `Connect to database` authority.

Refer to the DB2 documentation for more information about creating the database and users.

You must install the DB2 Administrative Client on the computer where you plan to run SDDBTool. See the following section for more information about SDDBTool.

2.2.4 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, make sure that Real-Time Decision Server is not running.

To initialize the Oracle RTD Database using SDDBTool:

1. Ensure that your database server is running.
2. If you are using Oracle Database as your Oracle RTD Database, 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

     Note: If you installed your Oracle RTD Database on a SQL Server named instance, specify Database Host\instance_name.

   - Database Port
   - Database Name (for Oracle Database, enter the SID instead of the 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.

     For Oracle Database, the Database Runtime User and the Database Administrative User are the same user.

5. Click **Next**.
6. Choose **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.

2.2.5 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 are three directories, **DB2Static**, **OracleStatic**, **SQLServerStatic**.

Each 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 appropriate `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 script appropriate for your database type:

- If you are using SQL Server for your Oracle RTD Database, run *RTD_HOME*`\examples\CrossSell\etc\data\SQLServer\initappdb.cmd`.
- If you are using Oracle Database for your Oracle RTD Database, run *RTD_HOME*`\examples\CrossSell\etc\data\Oracle\initappdb.cmd`.
- If you are using DB2 for your Oracle RTD Database, run *RTD_HOME*`\examples\CrossSell\etc\data\DB2\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. If you installed your Oracle RTD Database on a SQL Server named instance, specify *db_host\instance_name*.
db_port	The database port number.
db_name	The name of the database, or for Oracle Database, the SID.
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.

If you are using Oracle Database 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 script appropriate for your database type:

- If you are using SQL Server for your Oracle RTD Database, run *RTD_HOME*`\examples\DC_Demo\etc\data\SQLServer\initappdb.cmd`.
- If you are using Oracle Database for your Oracle RTD Database, run *RTD_HOME*`\examples\DC_Demo\etc\data\Oracle\initappdb.cmd`.
- If you are using DB2 for your Oracle RTD Database, run *RTD_HOME*`\examples\DC_Demo\etc\data\DB2\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 Quick Zip Installation

Oracle RTD Version 3.0.0.1 provides a fast way to set up and configure the Oracle RTD system. Oracle RTD Version 3.0.0.1 includes a zip file, **3.0_RTD_OC4J_QuickZip.zip**, that includes all the components required (except the database) to install Oracle RTD on a computer running Microsoft Windows.

The zip file includes a fully documented set of instructions, which describes the steps required to perform the installation. Make sure that you do not install into a path that contains spaces.

Caution: Because the quick zip installation installs standalone OC4J, which cannot be clustered, it is not recommended for production systems.

2.6 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.

2.6.1 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.

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

Configuring OC4J for Oracle Real-Time Decisions

Oracle RTD runs on the Oracle Containers for J2EE (OC4J) component of Oracle Application Server. You can choose to use either a standalone version of OC4J, or the version of OC4J that comes with Oracle Application Server.

Oracle RTD is supported on both UNIX and Windows platforms for OC4J. The following instructions describe how to install the Real-Time Decision Server on OC4J.

> **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.

This chapter contains the following topics:

- Section 3.1, "Configuring Server Properties"
- Section 3.2, "Providing JDBC Access to the Oracle RTD Database"
- Section 3.3, "Creating Oracle RTD Roles and Users"
- Section 3.4, "Installing the Oracle Real-Time Decisions Application on OC4J"
- Section 3.5, "Assigning Permissions to Custom Roles"
- Section 3.6, "Starting Oracle Real-Time Decisions"
- Section 3.7, "Uninstalling the Oracle Real-Time Decisions Application from OC4J"
- Section 3.8, "Configuring SSL for Real-Time Decision Server (Recommended)"
- Section 3.9, "Setting Up JConsole Security for OC4J"
- Section 3.10, "Changing the Oracle Real-Time Decisions Port Number in OC4J"
-

3.1 Configuring Server Properties

Perform the steps in the following sections to configure OC4J for Oracle RTD. Before you begin, make sure that OC4J is not running.

This section contains the following topics:

- Section 3.1.1, "Configuring Server Properties for Standalone OC4J"
- Section 3.1.2, "Configuring Server Properties for Oracle Application Server"

3.1.1 Configuring Server Properties for Standalone OC4J

Follow these steps to configure server properties for standalone OC4J:

1. Follow the steps appropriate to your operating system:

   - **On Windows:** Locate the file `OC4J_HOME\bin\oc4j.cmd` and open it for editing. After the line `:oc4j`, insert the following new lines. For *`jmx_remote_port`*, enter the value you want to use as the JConsole port number (for example, `12345`).

     ```
     set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote=true
     set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote.port=jmx_remote_port
     set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote.authenticate=false
     set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote.ssl=false
     set JVMARGS=%JVMARGS% -Djava.net.preferIPv4Stack=true
     set JVMARGS=%JVMARGS% -Xmx512M -XX:MaxPermsize=128M -server
     ```

     Tips: 1. If you get the error OutOfMemoryError, set the `-Xmx` argument to `-Xmx1024M`.

     2. If you get the error OutOfMemoryError:PermGen, set the `-XX:MaxPermSize` argument to `-XX:MaxPermSize=256M`.

     Then, save and close the file.

   - **On UNIX:** Locate the file `OC4J_HOME/bin/oc4j` and open it for editing. After the line `echo "Starting OC4J from $J2EE_HOME ..."`, insert the following new lines. For *`jmx_remote_port`*, enter the value you want to use as the JConsole port number (for example, `12345`).

     ```
     JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote=true
     JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.port=jmx_remote_port
     JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.authenticate=false
     JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.ssl=false
     JVMARGS=$JVMARGS -Djava.net.preferIPv4Stack=true
     JVMARGS=$JVMARGS -Xmx512M -XX:MaxPermsize=128M -server
     ```

     Tips: 1. If you get the error OutOfMemoryError, set the `-Xmx` argument to `-Xmx1024M`.

     2. If you get the error OutOfMemoryError:PermGen, set the `-XX:MaxPermSize` argument to `-XX:MaxPermSize=256M`.

     Then, save and close the file.

     Note: In order to perform this step, you may need to grant execute permission to the `OC4J_HOME/bin/oc4j` file.

2. Locate the file `OC4J_HOME/j2ee/home/config/default-web-site.xml` and copy it to the same location, renaming it to `rtd-web-site.xml`. In the renamed file, make the following changes:

   - In the `<web-site>` tag, set the `port` attribute to `8080`.
   - In the `<web-site>` tag, set the `protocol` attribute to `http`.
   - In the `<web-site>` tag, set the `display-name` attribute to `OC4J RTD Web Site`.
   - Delete all `<web-app>` tags.

- In the <access-log> tag, set the path attribute to ../log/rtd-web-access.log.

For example:

```
<?xml version="1.0"?>
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/web-site
-10_0.xsd"  port="8080" protocol="http" display-name="OC4J RTD Web Site"
schema-major-version="10" schema-minor-version="0" >
<default-web-app application="default" name="defaultWebApp" root="/j2ee" />
<access-log path="../log/rtd-web-access.log" split="day" />
</web-site>
```

Then, save and close the file.

3. Locate the file *OC4J_HOME*/j2ee/home/config/server.xml and open it for editing. After <web-site default="true" path="./default-web-site.xml" />, add the following line:

```
<web-site path="./rtd-web-site.xml" />
```

Then, save and close the file.

4. If you are using either SQL Server or DB2 for your Oracle RTD Database, copy the database JDBC jar files to the OC4J applib directory, as follows:

   - **For SQL Server:** Copy the file sqljdbc.jar in the directory *RTD_HOME*/lib/jdbc to the directory *OC4J_HOME*/j2ee/home/applib
   - **For DB2:** Copy the files db2jcc.jar and db2jcc_license_cu.jar in *RTD_HOME*/lib/jdbc to the directory *OC4J_HOME*/j2ee/home/applib

3.1.2 Configuring Server Properties for Oracle Application Server

Follow these steps to configure server properties for Oracle Application Server:

1. Locate the file *ORACLE_AS_HOME*/opmn/conf/opmn.xml and open it for editing. Add the following line to the end of the value attribute for the java-options tag, under the tags default_group, *oc4j_instance* (for example, home), and start-parameters:

```
-Dcom.sun.management.jmxremote=true -Dcom.sun.management.jmxremote.port=jmx_
remote_port -Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false -Djava.net.preferIPv4Stack=true
```

Make sure to enter this text as one continuous line. If you copy the text from this document, remove any line breaks that may appear. For *jmx_remote_port*, provide the value you want to use as the JConsole port number (for example, 12345).

For example:

```
<ias-component id="default_group">
  <process-type id="home" module-id="OC4J" status="enabled">
    <module-data>
      <category id="start-parameters">
        <data id="java-options" value="... -Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=12345
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-Djava.net.preferIPv4Stack=true"/>
```

> **Note:** In the preceding example, the `...` that appears before `-Dcom.sun.management.jmxremote=true` represents additional Java options parameters that are not shown in this example.

Then, save and close the file.

2. Locate the file *ORACLE_AS_HOME*/j2ee/*oc4j_instance*/config/default-web-site.xml and copy it to the same location, renaming it to rtd-web-site.xml.

   > **Note:** The value *oc4j_instance* refers to the particular instance of OC4J where you plan to deploy Oracle RTD. For example, if you plan to deploy Oracle RTD to the default OC4J instance, the path is: *ORACLE_AS_HOME*/j2ee/home/config/default-web-site.xml.

3. In the renamed file, make the following changes:

   - In the <web-site> tag, set the port attribute to 8080.
   - In the <web-site> tag, set the display-name attribute to OC4J RTD Web Site.
   - Delete all <web-app> tags.
   - In the <access-log> tag, set the path attribute to ../log/rtd-web-access.log.

   For example:

```
<?xml version="1.0"?>
<web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/web-site
-10_0.xsd"  port="8080" display-name="OC4J RTD Web Site"
schema-major-version="10" schema-minor-version="0" >
<default-web-app application="default" name="defaultWebApp" root="/j2ee" />
<access-log path="../log/rtd-web-access.log" split="day" />
</web-site>
```

   Then, save and close the file.

4. Locate the file *ORACLE_AS_HOME*/j2ee/oc4j_instance/config/server.xml and open it for editing. After `<web-site default="true" path="./default-web-site.xml" />`, add the following line:

```
<web-site path="./rtd-web-site.xml" />
```

   Then, save and close the file.

5. If you are using either SQL Server or DB2 for your Oracle RTD Database, copy the database JDBC jar files to the OC4J applib directory, as follows:

   - **For SQL Server:** Copy the file sqljdbc.jar in the directory *RTD_HOME*/lib/jdbc to the directory *ORACLE_AS_HOME*/j2ee/oc4j_instance/applib

- **For DB2:** Copy the files `db2jcc.jar` and `db2jcc_license_cu.jar` in *RTD_HOME*`/lib/jdbc` to the directory *ORACLE_AS_HOME*`/j2ee/oc4j_instance/applib`

3.2 Providing JDBC Access to the Oracle RTD Database

Use the OC4J management console, called the Application Server Control, to provide JDBC access to the Oracle RTD Database. For more information about how to use the Application Server Control, refer to the OC4J documentation.

This section describes how to create the required OC4J management elements, and how to combine the creation steps with the deployment of Oracle RTD on to OC4J.

The first step to perform is to create a shared library, as described in Section 3.2.1, "Creating the Shared Library oracle.rtd.jdbc."

After this, the sequence of steps to perform depends on the database type of your Oracle RTD Database:

- If the database is in SQL Server or DB2, the instructions to perform are in the following order::
  1. Section 3.2.2, "Creating a JDBC Provider for the Oracle RTD Database in SQLServer and DB2"
  2. Section 3.4, "Installing the Oracle Real-Time Decisions Application on OC4J"
- If the database is in Oracle, the instructions to perform are described in Section 3.2.3, "Creating a JDBC Provider for the Oracle RTD Database in Oracle."

This section consists of the folowing topics:

- Section 3.2.1, "Creating the Shared Library oracle.rtd.jdbc"
- Section 3.2.2, "Creating a JDBC Provider for the Oracle RTD Database in SQLServer and DB2"
- Section 3.2.3, "Creating a JDBC Provider for the Oracle RTD Database in Oracle"

3.2.1 Creating the Shared Library oracle.rtd.jdbc

Before you proceed, make sure that your OC4J server is not running.

To create the shared library **oracle.rtd.jdbc**, perform the following steps:

1. Start the OC4J server.

   If you are using the standalone version of OC4J, open a command prompt and go to *OC4J_HOME*`/bin`, then run the command `oc4j -start`.

   If you are using OC4J as part of Oracle Application Server, go to *ORACLE_AS_HOME*`/opmn/bin`, and use `opmnctl` to manage OC4J.

   See *Oracle Containers for J2EE Configuration and Administration Guide* for more information.

2. Log in to the Application Server Control as the `oc4jadmin` user. You can access the Application Server Control at `http://`*`oc4j_host`*`:`*`port`*`/em`. For the standalone version of OC4J, the port number is typically `8888`. For OC4J as part of Oracle Application Server, the port number is typically 80.

3. If you are using the standalone version of OC4J, go to the **Administration** tab, then click the Go to Task icon for the **Shared Libraries** task under the Properties heading.

If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Administration** tab.

4. On the Shared Libraries page, click **Create**, then enter the following data:
   - Shared Library Name = **oracle.rtd.jdbc**
   - Shared Library Version = **10.2.0.4**
5. Click **Next**.
6. On the Create Shared Libraries: Add Archives page, click **Add**, then Browse to select the file `RTD_HOME\lib\jdbc\ojdbc14.jar`.
7. Click **Continue**, then **Finish**.

3.2.2 Creating a JDBC Provider for the Oracle RTD Database in SQLServer and DB2

To create a JDBC provider for the Oracle RTD Database in SQL Server and DB2, perform the following steps:

1. If you have not already done so, log in to the OC4J Application Server Control, as described in steps 1 and 2 of Section 3.2.1, "Creating the Shared Library oracle.rtd.jdbc."
2. Go to the **Administration** tab, then click the **JDBC Resources** task under the Services heading. Under the Connection Pools heading, click **Create** to create a new connection pool for your Oracle RTD Database.

   If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Administration** tab.
3. On the Create Connection Pool - Application page, ensure that **default** is selected for **Application**, and that **New Connection Pool** is selected for **Connection Pool Type**. Then, click **Continue**.
4. On the Create Connection Pool page, enter `RTDConnectionPool` for **Name**.
5. For Connection Factory Class, enter one of the following:
   - **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`
   - **DB2:** `com.ibm.db2.jcc.DB2Driver`
6. For JDBC URL, enter one of the following:
   - **SQL Server:** `jdbc:sqlserver://`*`db_host`*`:`*`db_port`*`;databaseName=`*`db_name`*

     Note: If you installed your Oracle RTD Database on a SQL Server named instance, specify the *`db_host`* parameter using the format *`host_name\instance_name`*.
   - **DB2:** `jdbc:db2://`*`db_host`*`:`*`db_port`*`/`*`db_name`*

   `db_host` is the name of the server running the Oracle RTD Database, *`db_port`* is the port number for connecting to the database server, and *`db_name`* is the name of the Oracle RTD Database instance.
7. Under the Credentials heading, for **Username**, provide a database user name with system administration privileges for the Oracle RTD Database instance. Then, provide the corresponding password. See *Oracle Fusion Middleware Security Guide*

for information about whether to provide a cleartext password or an indirect password.

8. Click **Finish**.
9. On the JDBC Resources page, under the Data Sources heading, click **Create** to define a new data source.
10. On the Create Data Source - Application & Type page, ensure that **default** is selected for **Application**, and that **Managed Data Source** is selected for **Data Source Type**. Then, click **Continue**.
11. On the Create Data Source - Managed Data Source page, enter `RTD_DS` for **Name**, `jdbc/SDDS` for **JNDI Location**, and select **RTDConnectionPool** for **Connection Pool**. Keep the defaults for the other options. Then, click **Finish**.
12. On the JDBC Resources page, in the Data Sources table, click **Test Connection** for the **RTD_DS** data source. Follow these steps to test the RTD_DS data source:

    - Change the SQL Statement to `select * from SDAPPS`, then click **Test**.

    If the connection is not established successfully, restart OC4J and then test the data source again. If it still fails, ensure that your connection pool settings are correct.

After you have created and tested all the objects in Application Server Control, you must deploy and start Oracle RTD, by performing the instructions in Section 3.4, "Installing the Oracle Real-Time Decisions Application on OC4J" and Section 3.6, "Starting Oracle Real-Time Decisions."

3.2.3 Creating a JDBC Provider for the Oracle RTD Database in Oracle

To create a JDBC provider for the Oracle RTD Database in Oracle, you must first deploy OracleRTD on to OC4J, then set up the JDBC resources in Application Server Control, as follows:

1. If you have not already done so, start the OC4J server, as described in step 1 of Section 3.2.1, "Creating the Shared Library oracle.rtd.jdbc."
2. Deploy OracleRTD on to OC4J, by performing the instructions in Section 3.4, "Installing the Oracle Real-Time Decisions Application on OC4J."

    Ignore any errors about the missing Oracle RTD database SDDS.
3. In the Application Server Control, go to the **Administration** tab, then click the **JDBC Resources** task under the Services heading. Under the Connection Pools heading, click **Create** to create a new connection pool for your Oracle RTD Database.

    If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Administration** tab.
4. On the Create Connection Pool - Application page, ensure that **OracleRTD** (and *not* **default**), is selected for **Application**, and that **New Connection Pool** is selected for **Connection Pool Type**. Then, click **Continue**.
5. On the Create Connection Pool page, enter `RTDConnectionPool` for **Name**.
6. For Connection Factory Class, enter:

    - `oracle.jdbc.pool.OracleDataSource`
7. For JDBC URL, enter :

    - `jdbc:oracle:thin:@`*`db_host`*`:`*`db_port`*`:`*`sid`* *db_host* is the name of the server running the Oracle RTD Database, *db_port* is the port number for connecting to the database server, and *sid* is the Oracle System Identifier that refers to the instance of the Oracle Database running on the server, for example, ORCL.

8. Under the Credentials heading, for **Username**, provide a database user name with system administration privileges for the Oracle RTD Database instance. Then, provide the corresponding password. See *Oracle Fusion Middleware Security Guide* for information about whether to provide a cleartext password or an indirect password.
9. Click **Finish**.
10. On the JDBC Resources page, under the Data Sources heading, click **Create** to define a new data source.
11. On the Create Data Source - Application & Type page, ensure that **OracleRTD** (and *not* **default**), is selected for **Application**, and that **Managed Data Source** is selected for **Data Source Type**. Then, click **Continue**.
12. On the Create Data Source - Managed Data Source page, enter `RTD_DS` for **Name**, `jdbc/SDDS` for **JNDI Location**, and select `RTDConnectionPool for Connection Pool`. Keep the defaults for the other options. Then, click **Finish**.
13. On the JDBC Resources page, in the Data Sources table, click **Test Connection** for the **RTD_DS** data source, then click **Test**.

    If the connection is not established successfully, restart OC4J and then test the data source again. If it still fails, ensure that your connection pool settings are correct.

After you have created and tested all the objects in Application Server Control, do not redeploy OracleRTD. Start Oracle RTD as described in Section 3.6, "Starting Oracle Real-Time Decisions."

3.3 Creating Oracle RTD Roles and Users

> **Note:** For general information about roles and permissions in Oracle RTD, see Chapter 7, "Configuring Security for Oracle Real-Time Decisions."

The instructions here apply to both Oracle Application Server (Oracle AS) and to Oracle Container for J2EE (OC4J).

For the default security provider that comes with OC4J, the term "role" is the same as what other vendors might call a "user group". A role is a collection of users and other roles. Roles are created, and users added to roles, using OC4J's administration console.

Role Inheritance

OC4J has the concept of role inheritance. If role B inherits role A, then anyone in role B is also in role A. Using set theory as an analogy, B can be thought of as a subset completely contained with set A.

This section consists of the following topics:

- Section 3.3.1, "Creating Oracle RTD Roles"
- Section 3.3.2, "Creating Oracle RTD Administrators and Other Users"
- Section 3.3.3, "Updating Oracle RTD's Deployment Descriptors"

3.3.1 Creating Oracle RTD Roles

Table 3–1 shows the standard roles, two custom roles, and their inheritance relationships. All roles must inherit from RTDUsers, directly or indirectly.

Table 3–1 Standard Roles and Inherited Roles

Role	Inherited Roles	Standard or Custom?
RTDUsers	none	Standard
RTDAdministrators	RTDUsers	Standard
RTDDecisionCenterUsers	RTDUsers	Standard
RTDDecisionCenterEditors	RTDDecisionCenterUsers	Standard
RTDStudioDeployers	RTDUsers	Standard
RTDStudioDownloaders	RTDUsers	Standard
RTDBatchAdministrators	RTDUsers	Standard
RTDChoiceEditors	RTDUsers	Standard
ILS2Developers	RTDUsers	Custom
ILS2Users	RTDUsers	Custom

To create the standard roles, perform the following steps:

1. Open the OC4J administration console.
2. Click the Administration tab.
3. In the Task Name > Administration Tasks > Security > Security Providers line, click the **Go to Task** icon.
4. Click the **Instance Level Security** button.
5. Click the Realms tab.
6. In the Realm Name row for the default realm, **jazn.com**, click the number in the **Roles** column.
7. For each role in Table 3–1 to be created, perform the following steps:
    1. Click **Create**.
    2. Enter a name for the new role.

       Note: Do *not* select either of the following:

       - Grant RMI Login Permission
       - Grant Administration Permission

    3. Move any roles that should be inherited from the Available Roles list on the left to the Selected Roles list on the right.
    4. In the tree on the left, select **Security Realms**.
    5. In the window on the right, select the security realm to use.
    6. Click Apply.

3.3.2 Creating Oracle RTD Administrators and Other Users

To create a user, perform the following steps:

1. Open the OC4J administration console.
2. Click the Administration tab.
3. In the Task Name > Administration Tasks > Security > Security Providers line, click the **Go to Task** icon.
4. Click the **Instance Level Security** button.
5. Click the Realms tab.
6. In the Realm Name row for the default realm, **jazn.com**, click the number in the **Users** column.
7. For each user to be created, perform the following steps:
    1. Click **Create**.
    2. Enter a name for the new user.
    3. Enter a password for the user.
    4. Enter the password again, to confirm it.
    5. Move any roles for this user from the Available Roles list on the left to the Selected Roles list on the right.

       Note: Because of the way role inheritances are defined in Table 3–1, you do not need to explicitly add any users to the RTDUsers role.

    6. Click OK.

3.3.3 Updating Oracle RTD's Deployment Descriptors

Two of Oracle RTD's deployment descriptor files must be updated to reference any custom roles. These files are embedded within `ui.war` and `soap.war`, both of which are embedded within `RTD.ear`.

Note: This section describes the changes that need to be made to two `web.xml` files, but does not describe precisely how to find and update those files. The process is similar to the process of updating the `web.xml` files to contain custom database resource references, as described in Section 8.1.3, "Adding the New Data Source to Oracle RTD."

Update ui.war

The file represented by this path must be updated to reference any custom roles:

- rtd.ear/**ui.war**/web-inf/web.xml

There are three places within the file `web.xml` that need to be updated to reference custom roles. Look for the three occurrences of the string, "RTD-Custom-Roles", inside comments.

The first two places are for inserting security-role-ref elements, as in the following:

```
<!-- RTD-Custom-Roles. Insert custom RTD roles here, as might be needed for
```

```
         ILS-specific access. For example, RTD_ILS2Developers could be a role for
         deploying the Inline Service named ILS2, and RTD_ILS2Users for viewing or
         editing ILS2 in Decision Center.
 -->
      <security-role-ref>
          <role-name>RTD_ILS2Developers</role-name>
          <role-link>RTD_ILS2Developers</role-link>
      </security-role-ref>
      <security-role-ref>
          <role-name>RTD_ILS2Users</role-name>
          <role-link>RTD_ILS2Users</role-link>
      </security-role-ref>
```

The third place for custom roles is for inserting security-role elements, as in the following:

```
<!-- RTD-Custom-Roles. Insert here security-role elements for any custom RTD
roles,
         as might be needed for ILS-specific access. For example, RTD_
ILS2Developers
         could be a role for deploying the Inline Service named ILS2, and RTD_
ILS2Users
         for viewing or editing ILS2 in Decision Center.
-->
      <security-role>
          <role-name>RTD_ILS2Developers</role-name>
      </security-role>
      <security-role>
          <role-name>RTD_ILS2Users</role-name>
      </security-role>
```

Update soap.war

The file represented by this path must be updated to reference any custom roles:

- `rtd.ear/`**`soap.war`**`/web-inf/web.xml`

There are two places within the file `web.xml` that must be updated to reference custom roles. Look for the two occurrences of the string, "RTD-Custom-Roles", inside comments.

The first place is for inserting security-role-ref elements, as in the following:

```
 <!-- RTD-Custom-Roles. Insert custom RTD roles here, as might be needed for
         ILS-specific access. For example, RTD_ILS2Developers could be a role for
         deploying the Inline Service named ILS2, and RTD_ILS2Users for viewing or
         editing ILS2 in Decision Center.
-->
      <security-role-ref>
          <role-name>RTD_ILS2Developers</role-name>
          <role-link>RTD_ILS2Developers</role-link>
      </security-role-ref>
      <security-role-ref>
          <role-name>RTD_ILS2Users</role-name>
          <role-link>RTD_ILS2Users</role-link>
      </security-role-ref>
```

The second place for custom roles is for inserting security-role elements, as in the following:

```
<!-- RTD-Custom-Roles. Insert here security-role elements for any custom RTD
roles,
         as might be needed for ILS-specific access. For example, RTD_
ILS2Developers
         could be a role for deploying the Inline Service named ILS2, and RTD_
ILS2Users
         for viewing or editing ILS2 in Decision Center.
-->
      <security-role>
          <role-name>RTD_ILS2Developers</role-name>
      </security-role>
      <security-role>
          <role-name>RTD_ILS2Users</role-name>
      </security-role>
```

3.4 Installing the Oracle Real-Time Decisions Application on OC4J

Use the Application Server Control to install Oracle RTD on OC4J. Before you begin, ensure that OC4J is started.

To install Oracle RTD on OC4J:

1. If you have not already done so, log in to the Application Server Control as the `oc4jadmin` user. You can access the Application Server Control at `http://`*`oc4j_host`*`:`*`port`*`/em`. For the standalone version of OC4J, the port number is typically `8888`. For OC4J as part of Oracle Application Server, the port number is typically 80

2. If you are using the standalone version of OC4J, on the OC4J home page, click the **Applications** tab.

   If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Applications** tab.

3. Click **Deploy**. On the Deploy: Select Archive page, under the Archive heading, browse to specify the archive location *`RTD_HOME`*`/package/RTD.ear`. Then, under the Deployment Plan heading, ensure that **Automatically create a new deployment plan** is selected and click **Next**.

4. On the Deploy: Application Attributes page, enter `OracleRTD` for **Application Name**, then choose **rtd-web-site** for **Bind Web Module to Site**. Then, click **Next**.

5. On the Deploy: Deployment Settings page, click **Deploy**. It may take a few minutes for deployment to complete. Then, check to see if Oracle RTD is running by going to Decision Center at the URL `http://`*`server_name`*`:`*`port`*`/ui`. The port is typically `8080`.

3.5 Assigning Permissions to Custom Roles

As described in Section 7.4, "Assigning Permissions," assign Cluster permissions, Inline Service permissions, and Decision Center Perspective permissions to any custom roles.

3.6 Starting Oracle Real-Time Decisions

When you start OC4J, Oracle RTD is started automatically. To start Oracle RTD independently from OC4J, use the Application Server Control, as follows:

1. Log in to the Application Server Control as the `oc4jadmin` user. You can access the Application Server Control at `http://`*`oc4j_host`*`:`*`port`*`/em`.

2. Go to the **Applications** tab. In the Applications table, select **OracleRTD** and click **Start**. A green arrow pointing up in the Status column indicates that the application is started.

   If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Applications** tab.

3.7 Uninstalling the Oracle Real-Time Decisions Application from OC4J

You can use the Application Server Control to uninstall Oracle RTD from OC4J. Before you begin, ensure that OC4J is started.

To uninstall Oracle RTD from OC4J:

1. Log in to the Application Server Control as the `oc4jadmin` user. You can access the Application Server Control at `http://`*`oc4j_host`*`:`*`port`*`/em`.
2. Go to the **Applications** tab. In the Applications table, select **OracleRTD** and click **Stop**. A red arrow pointing down in the Status column indicates that the application is stopped.

   If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Applications** tab.
3. Click **Undeploy**, then click **Yes** on the Undeploy Application page.

These steps uninstall Oracle RTD from OC4J, but they do not remove the Oracle RTD files from the operating system. You must delete the Oracle RTD files manually.

3.8 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 2.6 to change the default Oracle RTD keystore password.

Note: If you want to use your own keystore, you do not need to complete the instructions in Section 2.6.

This section contains the following topics:

- Section 3.8.1, "Setting Up SSL"
- Section 3.8.2, "Testing the SSL Configuration"

3.8.1 Setting Up SSL

To configure SSL for Real-Time Decision Server

1. If you are using standalone OC4J, follow the instructions appropriate for your operating system. If you are using Oracle Application Server, proceed to Step 2.

   - **On Windows:** Locate the file *`OC4J_HOME`*`\bin\oc4j.cmd` and open it for editing. Before each `%JAVA_HOME%\bin\java` line, insert the following new line:

     ```
     cd %J2EE_HOME%
     ```

   - **On UNIX:** Locate the file *`OC4J_HOME`*`/bin/oc4j.sh` and open it for editing. Before each `$JAVA_HOME/bin/java` line, insert the following new line:

```
cd $J2EE_HOME
```

2. If you are using standalone OC4J, go to *OC4J_HOME*/`jlib`. If you are using Oracle Application Server, go to *ORACLE_AS_HOME*/`jlib`. Then, rename the file `jssl-1_1.jar` to `jssl-1_1.jar.notused`.

3. If you are using standalone OC4J, go to *OC4J_HOME*/`j2ee/home/config`. If you are using Oracle Application Server, go to *ORACLE_AS_HOME*/`j2ee/oc4j_instance/config`. Then, make a copy the of the file `rtd-web-site.xml` and rename the copied file to `rtd-secure-web-site.xml`.

4. Open `rtd-secure-web-site.xml` for editing and make the following changes:

   - In the `<web-site>` tag, set the `port` attribute to `8443`.
   - In the `<web-site>` tag, add or set the `secure` attribute to `true`.
   - In the `<web-site>` tag, set the `protocol` attribute to `https`.
   - In the `<web-site>` tag, set the `display-name` attribute to `OC4J RTD Secure Web Site`.
   - For each `<web-app>` tag, add or set the `shared` attribute to `true`.
   - In the `<access-log>` tag, set the `path` attribute to `../log/rtd-secure-web-access.log`.
   - Add the following line just before the `</web-site>` tag. Make sure to replace *RTD_HOME* with the actual Oracle RTD installation path.

     ```
     <ssl-config keystore="RTD_HOME/etc/ssl/sdserver.keystore"
     keystore-password="your_keystore_password" needs-client-auth="false" />
     ```

   For example:

   ```
   <?xml version="1.0" ?>
   <web-site xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/schema/web-site
   -10_0.xsd" port="8443" secure="true" protocol="https" display-name="OC4J Secure
   Default Web Site" schema-major-version="10" schema-minor-version="0" >
   <default-web-app application="default" name="defaultWebApp" />
   <web-app application="OracleRTD" name="ls" load-on-startup="true" root="/ls"
   shared="true" />
     <web-app application="OracleRTD" name="rtis" load-on-startup="true"
   root="/rtis" shared="true" />
     <web-app application="OracleRTD" name="soap" load-on-startup="true"
   root="/soap" shared="true" />
     <web-app application="OracleRTD" name="ui" load-on-startup="true" root="/ui"
   shared="true" />
     <web-app application="OracleRTD" name="schema" load-on-startup="true"
   root="/schema" shared="true" />
     <web-app application="OracleRTD" name="cactus" load-on-startup="true"
   root="/cactus" shared="true" />
     <access-log path="../log/rtd-secure-web-access.log" split="day" />
   <ssl-config keystore="C:\OracleBI\RTD\etc\ssl\sdserver.keystore"
   keystore-password="my_password" needs-client-auth="false" />
   </web-site>
   ```

   If you want to use your own keystore instead of using the default Oracle RTD keystore, provide the path to your own keystore in place of *RTD_HOME*/`etc/ssl/sdserver.keystore`.

5. Locate the file *OC4J_HOME*/j2ee/home/config/server.xml and open it for editing. After the line <web-site default="true" path="./default-web-site.xml" />, insert the following new line:

   ```
   <web-site path="./rtd-secure-web-site.xml" />
   ```

   Then, save and close the file.

6. Restart OC4J.

> **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 standalone OC4J or Oracle Application Server documentation for more information.

7. If you are using your own keystore and truststore, perform the following additional steps to enable SSL for Decision Studio and Load Generator. 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.

3.8.2 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. See:

`http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html` for instructions on generating a certificate request, and for importing the certificate from the CA into the keystore.

3.9 Setting Up JConsole Security for OC4J

Perform the following steps to set up security for the JConsole management tool. Before you begin, ensure that OC4J is not running.

To set up security for JConsole:

1. If you are using standalone OC4J, follow the steps appropriate to your operating system:

   - **On Windows:** Locate the file *`OC4J_HOME`*`\bin\oc4j.cmd` and open it for editing. Locate the line:

     ```
     set JVMARGS=%JVMARGS% -Dcom.sun.management.jmxremote.authenticate=false
     ```

     Change the value of this line to `true`. Then, save and close the file.

   - **On UNIX:** Locate the file *`OC4J_HOME`*`/bin/oc4j` and open it for editing. Locate the line:

     ```
     JVMARGS=$JVMARGS :-Dcom.sun.management.jmxremote.authenticate=false
     ```

     Change the value of this line to `true`. Then, save and close the file.

2. If you are using Oracle Application Server, locate the file *`ORACLE_AS_HOME`*`/opmn/conf/opmn.xml` and open it for editing. Locate the string `-Dcom.sun.management.jmxremote.authenticate=false`, then set the value to `true`. Then, save and close the file.

3. Locate the JDK install directory on the computer where OC4J is running. Be sure to go to the JDK installation that is being used by OC4J.

   - If you are using standalone OC4J, create a copy of the file *`JDK_HOME`*`/jre/lib/management/jmxremote.password.template` file and rename it to `jmxremote.password`. If you are unsure where your JDK is installed, check the `JAVA_HOME` operating system environment variable.

   - If you are using Oracle Application Server, create a copy of the file *`ORACLE_AS_HOME`*`/jdk/jre/lib/management/jmxremote.password.template` and rename it to `jmxremote.password`.

   Note: If Oracle Application Server is running on an AIX-based system, you downloaded the IBM Java 5 SR2 JDK, as described in Section 1.3.2. For this configuration, go to the IBM Java 5 SR2 installation directory, rather than *`ORACLE_AS_HOME`*. Then, navigate to the `jmx.remote.password.template` file.

4. 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.

5. 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.

6. 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://java.sun.com/j2se/1.5.0/docs/guide/management/security-windows.html`

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

   For more information on JConsole security, go to:

   `http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#PasswordAccess`

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

7. Start OC4J.

You can now run JConsole and log in, using the user name and password combination defined in the `jmxremote.password` file. See Section 15.1, "Accessing JConsole" for more information.

3.10 Changing the Oracle Real-Time Decisions Port Number in OC4J

To change the Oracle RTD application port number, you must change the port number in OC4J, as follows:

1. Stop OC4J.
2. If you are using standalone OC4J, locate the file *`OC4J_HOME`*`/j2ee/home/config/rtd-web-site.xml` and open it for editing. If you are using Oracle Application Server, locate the file *`ORACLE_AS_HOME`*`/j2ee/oc4j_instance/config/rtd-web-site.xml` and open it for editing.
3. In the `<web-site>` tag, set the `port` attribute to the new port number.

4. If you are using SSL for client connections to Oracle RTD, repeat Steps 2 and 3 for `rtd-secure-web-site.xml`.
5. Start OC4J.

4 Configuring WebSphere for Oracle Real-Time Decisions

Oracle RTD is supported on both UNIX and Windows platforms for IBM WebSphere application server. The following sections explain how to install the Real-Time Decision Server on WebSphere.

> **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.

This section contains the following topics:

4.1 Creating and Administrative User and Enabling Security

If you have already enabled security in WebSphere, you can skip this section. You should still check that Java 2 security is not turned on.

Use the WebSphere administrative console, called the Integrated Solutions Console, to enable security in WebSphere. For more information about how to use the Integrated Solutions Console, refer to the WebSphere documentation.

If security is disabled, follow these steps to enable security in WebSphere:

1. Access the Integrated Solutions Console at the URL `http://`*`websphere_host`*`:`*`port`*`/ibm/console`. At the login prompt, enter any user name. You will not need to enter any password. On Windows, you can also access the Integrated Solutions Console through **Start > Programs**.

   If you do not know the port number for the Integrated Solutions Console, you can find it in the `virtualhosts.xml` file, located in *`WEBSPHERE_HOME`*`/AppServer/profiles/`*`profile_name`*`/config/cells/`*`host_name`*.

2. In the tree on the left, expand **Security**, and choose **Secure administration, applications, and infrastructure**.
3. In the User account repository area, under the Available realm definitions heading, select **Federated repositories**, then click **Set as current**.
4. Click **Apply**, then **Save**.
5. Click **Configure**.
6. In the Federated repositories window, in the General Properties area, perform the following:

   a. Enter a **Realm Name**, such as RTDRealm.

   b. Enter a **Primary administrative user name**, such as admin.

   c. Under the Server user identity heading, select **Automatically generated server identity**.

7. In the Administrative user password window, in the General Properties area, enter the administrative user password in both the **Password** and **Confirm password** fields.
8. Click **OK**, then **Save**.
9. Log out, stop, then restart WebSphere.
10. Log in to the Integrated Solutions Console.

    If you do not know the port number for the Integrated Solutions Console, you can find it in the `virtualhosts.xml` file, located in *`WEBSPHERE_HOME`*`/AppServer/profiles/`*`profile_name`*`/config/cells/`*`host_name`*.

11. In the tree on the left, expand **Security** and choose **Secure administration, applications, and infrastructure**.
12. Under the Administrative security heading, select **Enable administrative security**.
13. Under the Application security heading, select **Enable application security**.
14. Under the Java 2 security heading, uncheck **Use Java 2 security to restrict application access to local resources**.
15. Click **Apply**, then **Save**.
16. Log out, stop, then restart WebSphere.

4.2 Configuring Server Properties

Use the WebSphere administrative console, called the Integrated Solutions Console, to configure server properties. For more information about how to use the Integrated Solutions Console, refer to the WebSphere documentation.

Follow these steps to configure server properties for WebSphere:

1. Start WebSphere. On Windows, you can use **Start > Programs** to start the server. On UNIX, go to `WEBSPHERE_HOME/AppServer/profiles/profile_name/bin` and run the following command:

   ```
   startServer.sh app_server_name -username admin_user -password admin_password
   ```

   For example:

   ```
   startServer.sh AppServer1 -username admin -password mypswd
   ```

2. 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**.

   If you do not know the port number for the Integrated Solutions Console, you can find it in the `virtualhosts.xml` file, located in `WEBSPHERE_HOME/AppServer/profiles/profile_name/config/cells/host_name`.

3. In the tree on the left, expand **Servers**, and choose **Application servers**.

4. Click the name of the application server where you want to run Oracle RTD (for example, **server1**). Under the Server Infrastructure heading, expand **Java and Process Management** and choose **Process Definition**.

5. Under the Additional Properties heading, click **Java Virtual Machine**.

6. Under the General Properties heading, in the **Generic JVM arguments** field, add the following string:

   ```
   -Djava.net.preferIPv4Stack=true
   ```

   If there is already a value in this field, add a space after the existing value, then add the new string.

7. Click **OK**.

   When you click **OK**, you may see an error stating that you need to provide values for **Initial Heap Size** and **Maximum Heap Size**. If you see this error, set these values as needed for your system (for example, you can set **Initial Heap Size** to `512` and **Maximum Heap Size** to `2048`).

8. On the **Java Virtual Machine** page, click **Custom Properties** under the Additional Properties heading.

9. Click **New**.

10. For **Name**, enter `org.eclipse.emf.ecore.EPackage.Registry.INSTANCE`.

11. For **Value**, enter `com.sigmadynamics.emf.util.SDEMFRegistry`.

12. Click **OK**, then click **Save**.

13. Follow these steps to add a custom property to adjust the JConsole MBean display:

    a. In the tree on the left, expand **Servers** and choose **Application servers**.

b. Click the name of the application server where you want to run Oracle RTD (for example, **server1**). Under the Server Infrastructure heading, expand **Administration** and choose **Administration Services**.

c. Under the Additional Properties heading, click **Custom Properties**, then click **New**.

d. For **Name**, enter `com.ibm.websphere.mbeans.disableRouting`.

e. For **Value**, enter `<on>OracleRTD:*</on>`.

f. Click **OK**, then click **Save**.

If you want to use the default HTTP transport port for your WebSphere profile as the default Oracle RTD application port, skip Steps 14 and 15 and proceed to Step 16. If you want to add a new HTTP transport port for Oracle RTD, continue at Step 14.

14. To add a new HTTP transport chain for Oracle RTD, follow these steps:

a. In the tree on the left, expand **Servers** and choose **Application servers**.

b. Click the name of the application server where you want to run Oracle RTD (for example, **server1**). Under the Container Settings heading, expand **Web container settings** and choose **Web container transport chains**.

c. Click **New**.

d. For **Transport chain name**, enter `OracleRTD_chain`.

e. For **Transport chain template**, select **WebContainer(templates/chains | webcontainer-chains.xml#Chain_1)**. Then, click **Next**.

f. For **Port name**, enter `OracleRTD_port` or some similar value.

g. For **Host**, keep the default value, `*`.

h. For **Port**, enter the port number you want to use for Oracle RTD. By default, the Oracle RTD application runs on port `8080`.

i. Click **Next**, then click **Finish**, then click **Save**.

15. If you added a new HTTP transport port in Step 14, follow these steps to create a new host alias. Otherwise, proceed to Step 16.

a. In the tree on the left, expand **Environment** and choose **Virtual Hosts**.

b. Click **default_host**. Under the Additional Properties heading, click **Host Aliases**.

c. Click **New**. Keep the default value, `*`, for **Host**, then enter the Oracle RTD port number for **Port**.

d. Click **OK**, then click **Save**.

16. Restart WebSphere.

4.3 Creating Oracle RTD Roles and Users

In WebSphere, 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 map the roles to groups. Users may be assigned to or removed from the groups at any time to add them or remove them from the referencing roles.

This section consists of the following topics:

- Section 4.3.1, "Creating Users for Oracle RTD"
- Section 4.3.2, "Creating Groups"
- Section 4.3.3, "Standard Oracle RTD Roles"

4.3.1 Creating Users for Oracle RTD

To create users, perform the following steps:

1. Log into the Integrated Solutions Console, using the administrative user and password.
2. In the tree on the left, expand **Users and Groups**, and choose **Manage Users.**
3. Click the **Create...** button.
4. Enter user and password information for the user, such as rtdadmin for the User ID.

   You will use the User ID later on, when you add the user to one or more groups.
5. Click **Create**, then **Close**.
6. Repeat steps 3 to 5 to create other users.

4.3.2 Creating Groups

To create groups, perform the following steps:

1. Log into the Integrated Solutions Console, using the administrative user and password.
2. In the tree on the left, under **Users and Groups**, choose **Manage Groups.**
3. Click the **Create...** button.
4. In the Group name field, enter RTDAdminGroup.
5. Click **Create**, then **Close**.
6. In the Manage Groups page, click the group name RTDAdminGroup.
7. Click the **Add Users...** button.
8. Click **Search** to display a list of users.
9. In the search result list, select the user name to add to the RTDAdminGroup.
10. Click **Add**, then **Close**.
11. Repeat steps 2 through 10 to create each of the following groups for Oracle RTD:
    - RTDDCEditorGroup
    - RTDDCUserGroup
    - RTDStudioDeployerGroup
    - RTDStudioDownloaderGroup
    - RTDBatchAdminGroup
    - RTDChoiceEditorGroup
    - RTDUserGroup
12. In the Manage Groups page, click the group name RTDUserGroup.
13. In the Group Properties area, click the Members tab.

14. Click the **Add Groups...** button.
15. Add all the groups created for Oracle RTD except RTDUserGroup.
16. Click **Close**.

4.3.3 Standard Oracle RTD Roles

The groups specified in Section 4.3.2, "Creating Groups" are automatically mapped to the standard Oracle RTD roles, as shown in Table 4–1.

Table 4–1 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

Section 7.2, "Standard Oracle RTD Roles" of *Oracle Real-Time Decisions Installation and Administration Guide* 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.

> **Note:** For information about custom roles, see Section 4.9, "Creating Custom Roles and Assigning Permissions to Custom Roles (Optional)."

4.4 Creating a JDBC Provider for the Oracle RTD Database

Use the Integrated Solutions Console to create a JDBC provider for the Oracle RTD Database. Before you begin, ensure that WebSphere is started.

To create a JDBC provider for the Oracle RTD Database:

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 **Resources**, then expand **JDBC** and choose **JDBC Providers**.
3. Ensure that the scope is set correctly (to **Node=*host_name*Node_*number***), then click **New**.
4. For **Database type**, select **User-defined**.
5. For **Implementation class name**, enter one of the following values:

   - **For SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`
   - **For Oracle Database:** `oracle.jdbc.pool.OracleConnectionPoolDataSource`
   - **For DB2:** `com.ibm.db2.jcc.DB2ConnectionPoolDataSource`

6. For **Name**, enter `RTDDataProvider`, then click **Next**.
7. For Class path, enter one of the following values:
   - **For SQL Server:** *RTD_HOME*`/lib/jdbc/sqljdbc.jar`
   - **For Oracle Database:** *RTD_HOME*`/lib/jdbc/ojdbc14.jar`
   - **For DB2:** *RTD_HOME*`/lib/jdbc/db2jcc.jar;`*RTD_HOME*`/lib/jdbc/db2jcc_license_cu.jar`
8. Click **Next**, then click **Finish**, then click **Save**.
9. On the JDBC Providers page, click **RTDDataProvider**. Then, under the Additional Properties heading, click **Data sources**.
10. Click **New**, then, under the Component-managed authentication alias and XA authentication alias heading, click **create a new J2C authentication alias**.
11. Click **New**, then provide the following values:

    a. For **Alias**, enter `RTDDS_auth`.

    b. For **User ID**, enter the name of the database run-time user.

    c. For **Password**, enter the corresponding password for the database user. The password cannot be blank.

    After you have entered these values, click **OK**, then click **Save**.
12. Return to the Data sources page using the locator link, then click **New**. Provide the following values:

    a. For **Data source name**, enter `RTD_DS`, or a similar value.

    b. For JNDI name, enter `SDDS`. The JNDI name *must* be `SDDS` for Oracle RTD to access the database.

    c. For **Component-managed authentication alias and XA authentication**, select `RTDDS_auth`.

    After you have entered these values, click **Next**.
13. On the **Enter database specific properties for the data source** page, click Next.
14. On the Summary page, click **Finish**, then click **Save**.
15. Click the name of the data source you created (for example, **RTD_DS**).
16. If you are using SQL Server for your Oracle RTD Database, follow these steps to set properties for the data source:

    a. Under the Additional Properties heading, click **Custom properties**.

    b. Click the **Select All** icon, then click **Delete** to delete any existing custom properties.

    c. Click **New**, enter `databaseName` for **Name**, then enter the name of your database for **Value**. Then, click **OK**.

   d. Click **New**, enter `portNumber` for **Name**, then enter the port number of your database (typically `1433`) for **Value**. Then, click **OK**.

   e. Click **New**, enter `serverName` for **Name**, then enter the name of your database server for **Value**.

   > **Note:** If you installed your Oracle RTD Database on a SQL Server named instance, specify the name of your database server using the format *`host_name`*`\`*`instance_name`*.

   f. Click **OK**.

   g. Click **Save**.

17. If you are using Oracle Database for your Oracle RTD Database, follow these steps to set properties for the data source:

   a. Scroll to the bottom of the page and locate the Oracle data source properties section.

   b. For **URL**, enter `jdbc:oracle:thin:@`*`db_host`*`:`*`db_port`*`:`*`sid`*.

   For example: `jdbc:oracle:thin:@dbhost.company.com:1521:orcl`

   c. Click **OK**, then click **Save**.

18. If you are using DB2 for your Oracle RTD Database, follow these steps to set properties for the data source:

   a. Scroll to the bottom of the page and locate the DB2 Universal data source properties section.

   b. For **Database name**, enter the name of your database.

   c. For **Driver type**, enter `4`.

   d. For **Server name**, enter the name of your database server.

   e. For **Port number**, enter the port number of your database.

   f. Click **OK**, then click **Save**.

19. Restart WebSphere, then launch the Integrated Solutions Console again.

20. Expand **Resources**, then expand **JDBC** and choose **Data Sources**.

21. Select **RTD_DS** and click **Test Connection**. If the connection fails, ensure that your data source settings are correct, then test the data source again.

4.5 Installing the Oracle Real-Time Decisions Application on WebSphere

Use the Integrated Solutions Console to install Oracle RTD on WebSphere. Before you begin, ensure that WebSphere is started.

To install Oracle RTD on WebSphere:

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 **Applications**, then choose **Enterprise Applications**.

3. Remove any sample applications, such as `ivtApp` or `query`. To do this, follow these steps:

   a. Select the sample application.

   b. Click **Stop**.

   c. Click **Uninstall**.

   d. Click **Save**.

4. Click **Install**.

5. In the Path to the new application section, enter or browse to the path *`RTD_HOME`*`/package/RTD.ear`. Make sure to replace *`RTD_HOME`* with the actual Oracle RTD installation path.

6. Click **Next**, then click **Next** again, then click **Next** again.

7. Click **Finish**, then click **Save**.

8. Click the Enterprise Application Name **OracleRTD**.

9. In the References area, select **Resource references**.

10. Under the Specify authentication method heading, select **Use default method (many-to-one mapping)**.

11. Select the J2C authentication **RTDDS_auth** that you created in step 11 of Section 4.4, "Creating a JDBC Provider for the Oracle RTD Database."

12. Click the **Select All** icon, then click **Apply**.

13. For each module, set the **Target Resource JNDI Name** to `SDDS`.

14. Click **OK**, then **Save**.

4.6 Starting Oracle Real-Time Decisions

After you install Oracle RTD on WebSphere, Oracle RTD is not started by default. You can use the Integrated Solutions Console to start Oracle RTD. Before you begin, ensure that WebSphere is started.

To start Oracle RTD:

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 **Applications** and click **Enterprise Applications**.

3. Select **OracleRTD** and click **Start**. Oracle RTD may take a few minutes to start up.

4. Check to see if Oracle RTD is running by going to Decision Center at the URL `http://`*`server_name`*`:`*`port`*`/ui`.

After you start Oracle RTD for the first time, the application will be started and stopped automatically when you start and stop WebSphere. To start and stop Oracle RTD independently from WebSphere, use the Integrated Solutions Console.

4.7 Setting Classloader Priority

To give priority to the application classloader over the application server classloader, after you have deployed Oracle RTD, you must explicitly set the Classloader priority in the Websphere administrative console, as follows:

1. Log into the WebSphere administrative console, using the administrative user and password.
2. Expand **Applications**, then choose **Enterprise Applications**.
3. Click **OracleRTD**.
4. Click **Class loading and update detection**.
5. Select **Classes loaded with application class loader first**.
6. Click OK.
7. Restart WebSphere.

4.8 Viewing and Changing User-Role Associations

The standard Oracle RTD Roles and Roles-groups mapping have been predefined for WebSphere.

To show the Oracle RTD Roles and Roles-groups mapping, perform the following steps:

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**.

If you want to view or edit the user-role associations, check the role to be modified, then click either **Look up users** or **Look up groups** and change the mapping.

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

This section consists of the following topics:

- Section 4.9.1, "Creating Custom Roles"
- Section 4.9.2, "Assigning Permissions to Custom Roles"

4.9.1 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 4.3.2, "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 4.9.1.1, "Specifying Roles in application.xml" for details.
3. Map the roles to the WebSphere groups in the Integrated Solutions Console. See Section 4.9.1.2, "Mapping Roles to Groups" for details.

Then, perform either of the following:

1. Uninstall, then redeploy Oracle RTD, as follows:
   a. Download the two deployment descriptor files, `application.xml` and `ibm-application-bnd.xmi`, back into `RTD.ear` (the file `ibm-application-bnd.xmi` contains the role-to-group mappings).
   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 4.9.1.1, "Specifying Roles in application.xml"
- Section 4.9.1.2, "Mapping Roles to Groups"

4.9.1.1 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 id="SecurityRole_1241469153092">
        <role-name>ILS2Users</role-name>
    </security-role>
```

   where `security-role id` is any unique value.

2. Repeat step 1 for as many roles as you want to create.

4.9.1.2 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.

After you have finished mapping the roles to groups in the Integrated Solutions Console, your changes are saved in the deployment descriptor file `ibm-application-bnd.xmi`.

> **Note:** To view and change the user-role associations, see Section 4.8, "Viewing and Changing User-Role Associations."

4.9.2 Assigning Permissions to Custom Roles

As described in Section 7.4, "Assigning Permissions" of *Oracle Real-Time Decisions Installation and Administration Guide*, assign Cluster permissions, Inline Service permissions, and Decision Center Perspective permissions to any custom roles.

4.10 Uninstalling the Oracle Real-Time Decisions Application from WebSphere

You can use the Integrated Solutions Console to uninstall Oracle RTD from WebSphere. Before you begin, ensure that WebSphere is started.

To uninstall Oracle RTD from WebSphere:

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 **Applications**, then choose **Enterprise Applications**.
3. Select **OracleRTD** and click **Stop**.
4. Click **Uninstall**, then click **Save**.

These steps uninstall Oracle RTD from WebSphere, but they do not remove the Oracle RTD files from the operating system. You must delete the Oracle RTD files manually.

4.11 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 2.6, "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 2.6.

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:
>
> 1. 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 Center and Load Generator. 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.

4.11.1 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 the following URL:

`http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html`

4.12 Setting Up JConsole for WebSphere

This section provides information about setting up the JConsole management tool.

This section contains the following topics:

- Section 4.12.1, "Determining the WebSphere Bootstrap Port Number"
- Section 4.12.2, "Setting Up a Batch File for JConsole"
- Section 4.12.3, "Creating a JConsole User"

4.12.1 Determining the WebSphere Bootstrap Port Number

The WebSphere bootstrap port number is used as the JMX remote port, used for accessing JConsole. You need to know the port value to set up JConsole access.

Follow these steps to find the value of the bootstrap port number, and to change it if necessary:

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 **Servers** and choose **Application servers**.
3. Click the name of the application server where Oracle RTD is running (for example, **server1**). Under the Communications heading, click **Ports**.
4. You can find the value of the bootstrap port by looking at the **BOOTSTRAP_ADDRESS** entry in the Ports table. You can note this value and use it for the JConsole port in subsequent sections, or you can change the bootstrap port number. To change the bootstrap port, follow these steps:

   a. Click **BOOTSTRAP_ADDRESS**.

   b. For **Port**, enter the port number you want to use for JConsole (for example, `12345`).

   c. Click **OK**, then click **Save**.

   d. Restart WebSphere.

4.12.2 Setting Up a Batch File for JConsole

Follow the instructions appropriate for your operating system:

- Section 4.12.2.1, "Setting Up a JConsole Batch File for Windows"
- Section 4.12.2.2, "Setting Up a JConsole Batch File for Linux or AIX-Based Systems"
- Section 4.12.2.3, "Setting Up a JConsole Batch File for Solaris"

4.12.2.1 Setting Up a JConsole Batch File for Windows

On Windows operating systems, create a batch file named `startJConsole.bat` and include the following:

```
set WAS_HOME=WEBSPHERE_HOME\AppServer
set USER_HOME=WEBSPHERE_HOME\AppServer\profiles\profile_name
set WAS_HOST=localhost
set WAS_BOOTSTRAP_PORT=jmx_remote_port

"%WAS_HOME%\java\bin\jconsole" -J-Djava.class.path="%WAS_
HOME%\runtimes\com.ibm.ws.admin.client_6.1.0.jar;%WAS_
HOME%\java\lib\tools.jar" -J-Dcom.ibm.CORBA.ConfigURL="file:%USER_
HOME%\properties\sas.client.props" -J-Dcom.ibm.SSL.ConfigURL="file:%USER_
HOME%\properties\ssl.client.props" service:jmx:iiop://%WAS_HOST%:%WAS_
BOOTSTRAP_PORT%/jndi/JMXConnector
```

For *`jmx_remote_port`*, enter the JConsole port number. For WebSphere, the JMX remote port is always the same as the WebSphere bootstrap port. Make sure to replace *`WEBSPHERE_HOME`* with the actual WebSphere installation path, and replace *`profile_name`* with the name of your WebSphere profile. For example:

```
set WAS_HOME=C:\Program Files\IBM\WebSphere\AppServer
set USER_HOME=C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01
set WAS_HOST=localhost
set WAS_BOOTSTRAP_PORT=12345

"%WAS_HOME%\java\bin\jconsole" -J-Djava.class.path="%WAS_
HOME%\runtimes\com.ibm.ws.admin.client_6.1.0.jar;%WAS_
HOME%\java\lib\tools.jar" -J-Dcom.ibm.CORBA.ConfigURL="file:%USER_
HOME%\properties\sas.client.props" -J-Dcom.ibm.SSL.ConfigURL="file:%USER_
HOME%\properties\ssl.client.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.

4.12.2.2 Setting Up a JConsole Batch File for Linux or AIX-Based Systems

On Linux or AIX-based systems, create a shell script named `startJConsole.sh` and include the following:

```
#!/bin/sh
WAS_HOME=WEBSPHERE_HOME/AppServer
USER_HOME=WEBSPHERE_HOME/AppServer/profiles/profile_name
WAS_HOST=localhost
WAS_BOOTSTRAP_PORT=jmx_remote_port

$WAS_HOME/java/bin/jconsole -J-Djava.class.path=$WAS_HOME/runtimes/com.ibm.
ws.admin.client_6.1.0.jar:$WAS_HOME/java/lib/tools.jar -J-Dcom.ibm.CORBA.
ConfigURL=file:$USER_HOME/properties/sas.client.props -J-Dcom.ibm.SSL.
ConfigURL=file:$USER_HOME/properties/ssl.client.props service:jmx:iiop://$WAS_
HOST:$WAS_BOOTSTRAP_PORT/jndi/JMXConnector
```

For *jmx_remote_port*, enter the JConsole port number (for example, 12345). For WebSphere, the JMX remote port is always the same as the WebSphere bootstrap port. Make sure to replace *WEBSPHERE_HOME* with the actual WebSphere installation path, and replace *profile_name* with the name of your WebSphere profile.

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

4.12.2.3 Setting Up a JConsole Batch File for Solaris

On Solaris operating systems, create a shell script named `startJConsole.sh` and include the following:

```
#!/bin/sh
WAS_HOME=WEBSPHERE_HOME/AppServer
USER_HOME=WEBSPHERE_HOME/AppServer/profiles/profile_name
WAS_HOST=localhost
WAS_BOOTSTRAP_PORT=jmx_remote_port

$WAS_HOME/java/bin/jconsole -J-Djava.class.path=$WAS_HOME/runtimes/com.ibm.ws.
admin.client_6.1.0.jar:$WAS_HOME/java/lib/tools.jar:$WAS_HOME/java/lib/jconsole.
jar -J-Dcom.ibm.CORBA.ConfigURL=file:$USER_HOME/properties/sas.client.
props -J-Dcom.ibm.SSL.ConfigURL=file:$USER_HOME/properties/ssl.client.props
service:jmx:iiop://$WAS_HOST:$WAS_BOOTSTRAP_PORT/jndi/JMXConnector
```

For *jmx_remote_port*, enter the JConsole port number (for example, 12345). For WebSphere, the JMX remote port is always the same as the WebSphere bootstrap port. Make sure to replace *WEBSPHERE_HOME* with the actual WebSphere installation path, and replace *profile_name* with the name of your WebSphere profile.

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

4.12.3 Creating a JConsole User

Follow these steps to create a JConsole user:

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 **Users and Groups** and choose **Manage Users**.
3. Click **Create**, then provide information for the JConsole user you want to create. For example, you could enter `jmx_admin` for **User ID**, `JMX` for **First name**, `Admin` for **Last name**, then provide a password.
4. Click **Create**.
5. In the tree on the left, expand **Users and Groups** and choose **Administrative User Roles**.
6. Click **Add**.
7. For **User**, enter the User ID you provided in Step 3.
8. For **Role(s)**, select **Administrator**.
9. Click **OK**, then click **Save**.

You can now run JConsole and log in, using the User ID and password you just created. See Section 15.1, "Accessing JConsole" for more information.

4.13 Changing the Oracle Real-Time Decisions Port Number in WebSphere

To change the Oracle RTD application port number in WebSphere, perform the following steps:

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 **Servers** and choose **Application server**.
3. Click the name of the application server where Oracle RTD is running (for example, **server1**). Under the Container Settings heading, expand **Web container settings** and choose **Web container transport chains**.
4. Click the name of the transport chain that corresponds to the port number you want to change. For example, if you set up a new transport chain for Oracle RTD, click **OracleRTD_chain**. If the Oracle RTD application is using the default transport chain for this application server profile, click **WCInboundDefault**.
5. Click the TCP inbound channel link (for example, **TCP inbound channel (TCP 6)**).
6. Under the Related Items heading, click **Ports**.
7. Click the name of the port number you want to change (for example, **OracleRTD_port** or **WC_defaulthost**).
8. For **Port**, enter the new port number you want to use for Oracle RTD.
9. Click **OK**, then click **Save**.
10. In the tree on the left, expand **Environment** and choose **Virtual Hosts**.
11. Click **default_host**. Under the Additional Properties heading, click **Host Aliases**.
12. Click the * link that corresponds to the previous value of the Oracle RTD port.
13. For **Port**, enter the new port number you want to use for Oracle RTD.

14. Click **OK**, then click **Save**.
15. Restart WebSphere.

5 Configuring WebLogic for Oracle Real-Time Decisions

Oracle RTD is supported on both UNIX and Windows platforms for the Oracle WebLogic application server. The following sections explain how to install 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.

This section contains the following topics:

5.1 Creating and Starting an Oracle RTD Domain

To create the domain for Oracle RTD, perform the following steps:

1. Start the Configuration Wizard:

   Start > All Programs > Oracle WebLogic > WebLogic Server > Tools > Configuration Wizard.

2. In the Welcome dialog, select the **Create a new WebLogic Domain** radio button.
3. Click **Next**.
4. In the Domain Source dialog, select **Generate a domain configured automatically to support the following products** radio button.
5. Click **Next**.
6. In the Configure Administration Username and Password dialog, enter admin username/password, for example weblogic/weblogic.
7. Click **Next**.
8. In the Configure Server Start Mode and JDK dialog, under WebLogic Domain Startup Mode, select the appropriate development mode. Note that for Production mode, you will need to use Edit & Lock key for changes.
9. In the Configure Server Start Mode and JDK dialog, under JDK Selection, select **Available JDKs** radio button, then select a JDK that is supported for your system configuration, for example, Sun, JRockit.
10. Click **Next**.
11. In the Customize Environment and Services Settings dialog, select **No**.
12. Click **Next**.
13. In the Create WebLogic Domain dialog, for Domain Name, enter a name for the Oracle RTD domain, such as `RTD_domain` and click **Create**.
14. Wait until the installation finishes.
15. Check the **Start Admin Server** checkbox.
16. Click **Done**.
17. Stop the Example domain:

    Start > All Programs > Oracle WebLogic > WebLogic Server > Examples > Stop Examples Server

18. Start the domain:

    Start > All Programs > Oracle WebLogic > User Projects > *domain_name* > Start Admin Server for Weblogic Server Domain

5.2 Starting the Node Manager and the WebLogic Server

To start the Node Manager and the WebLogic Server, perform the following steps:

1. Navigate the path:

   Start > All Programs > Oracle WebLogic > WebLogic Server > Tools > Node Manager

2. Start the WebLogic Server:

   In Windows, navigate the path:

   Start > All Programs > Oracle WebLogic > QuickStart

5.3 Creating a Managed Server

To create a managed server, perform the following steps:

1. Open the Administration Console for your Oracle RTD domain.

   `http://`*`weblogic_host`*`:`*`port`*`/console`

   For example, `http://localhost:7001/console`.

2. For the WebLogic Server, enter the Username and Password for the domain, and click Log In.
3. In the tree on the left, expand Environment and choose Servers.
4. Add a new server, and specify the Server Name, for example, RTD_Server.
5. For the Server Listen Port, enter the port number that you want to use for Oracle RTD, for example, 8080.
6. Click Finish.

5.4 Configuring Managed Server Properties

To add JVM and startup arguments to the managed server, perform the following steps:

1. Navigate the path:

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

2. For **Java Home** and **Java Vendor**, enter the location and vendor name of the JDK that is supported for your system configuration.

   For example, `C:\Oracle\Middleware\jdk160_23` and **Sun**, or `C:\Oracle\Middleware\jrockit_1.6.0_20` and **BEA**.

3. Add the appropriate path for your configuration to **ClassPath**:

   On Windows

   - For Oracle

     ```
     MW_HOME\wlserver_10.3\server\lib\weblogic.jar;RTD_
     HOME\lib\jdbc\ojdbc14.jar
     ```

   - For SQLServer

     ```
     MW_HOME\wlserver_10.3\server\lib\weblogic.jar;RTD_
     HOME\lib\jdbc\sqljdbc.jar
     ```

   - For DB2

     ```
     MW_HOME\wlserver_10.3\server\lib\weblogic.jar;RTD_
     HOME\lib\jdbc\db2jcc.jar;RTD_HOME\lib\jdbc\db2jcc_license_
     cu.jar
     ```

   On Linux/Unix

   - For Oracle

     ```
     MW_HOME\wlserver_10.3\server\lib\weblogic.jar:RTD_
     HOME\lib\jdbc\ojdbc14.jar
     ```

   - For SQLServer

     ```
     MW_HOME\wlserver_10.3\server\lib\weblogic.jar:RTD_
     HOME\lib\jdbc\sqljdbc.jar
     ```

- For DB2

  `MW_HOME\wlserver_10.3\server\lib\weblogic.jar:RTD_HOME\lib\jdbc\db2jcc.jar:RTD_HOME\lib\jdbc\db2jcc_license_cu.jar`

4. For the Arguments field, add the following, all on one line:

```
-Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.S
DEMFRegistry
-Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SA
XParserFactoryImpl -Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=12345
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false -Djava.net.preferIPv4Stack=true
-Dweblogic.wsee.skip.async.response=true -XX:MaxPermSize=256m
```

5. Save.

5.5 Creating and Adding a Machine to the Managed Server

To create a machine and add it to the managed server, 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, expand **Environment** and click **Machines**.
3. Click **New**.
4. For **Name**, enter the WebLogic server host name, for example, My`Machine`.
5. Click **Save**.
6. Under Environments > Machines > <*your_newly_created_machine*>, click the Servers tab, then **Add**.
7. In the Add A Server to the Machine page, select the server that you created in step 4 of Section 5.3, "Creating a Managed Server" from the dropdown list.
8. Click **Finish**.

5.6 Starting the Managed Server

Note: The instructions in the main part of this section are for Oracle RTD deployed onto WebLogic managed servers when they are started and stopped through node manager.

For the WebLogic server startup instructions where you deploy Oracle RTD onto the WebLogic admin server, or where you start managed servers without going through node manager, see Section 5.6.1, "Starting Oracle RTD When Deployed to Admin Server or Through Startup Scripts."

To start the managed server, 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, expand **Environment** and click **Servers**.
3. Click the **Control** tab.
4. Select the checkbox for the managed server that you created in Section 5.3, "Creating a Managed Server.", and start the server.
5. Wait a few seconds before refreshing the page.

   The server that you selected should be in a Running state.

5.6.1 Starting Oracle RTD When Deployed to Admin Server or Through Startup Scripts

If you deploy Oracle RTD onto the WebLogic admin server, or if you start managed servers without going through node manager, you must specify the JVM arguments as detailed in step 4 of Section 5.4, "Configuring Managed Server Properties" in one of two ways:

1. Update the appropriate startup scripts:

- `startWebLogic.cmd` and `startWebLogic.sh` for admin server
- `startManagedWebLogic.cmd` and `startManagedWebLogic.sh` for managed servers

by adding the JVM arguments that follow to the top of the script, with one line for each JVM argument.

For example, for **Windows** *(extra blank lines are shown for clarity, but are not required in the script*):

```
set JAVA_OPTIONS=%JAVA_OPTIONS%
-Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.SDEM
FRegistry

set JAVA_OPTIONS=%JAVA_OPTIONS%
-Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXPa
rserFactoryImpl

set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote=true
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.port=12345
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false
set JAVA_OPTS=%JAVA_OPTS% -Djava.net.preferIPv4Stack=true
set JAVA_OPTS=%JAVA_OPTS% -Dweblogic.wsee.skip.async.response=true
set JAVA_OPTS=%JAVA_OPTS% -XX:MaxPermSize=128m
```

For example, for **Unix** *(extra blank lines are shown for clarity, but are not required in the script*):

```
JVMARGS=$JVMARGS
-Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.SDEM
FRegistry

JVMARGS=$JVMARGS
-Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXPa
rserFactoryImpl

JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote=true
JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.port=12345
JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.authenticate=false
JVMARGS=$JVMARGS -Dcom.sun.management.jmxremote.ssl=false
```

```
JVMARGS=$JVMARGS -Djava.net.preferIPv4Stack=true
JVMARGS=$JVMARGS -Dweblogic.wsee.skip.async.response=true
JVMARGS=$JVMARGS -XX:MaxPermSize=128m
```

2. When calling the scripts to start the admin server or managed servers, pass in the JVM arguments.

For example:

```
>startManagedWebLogic.sh Managed_Server_Name http://${IP_ADDRESS}:${MS_PORT}
-Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.SDEM
FRegistry
-Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXPa
rserFactoryImpl -Dcom.sun.management.jmxremote=true
-Dcom.sun.management.jmxremote.port=12345
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false -Djava.net.preferIPv4Stack=true
-Dweblogic.wsee.skip.async.response=true -XX:MaxPermSize=128m
```

Note: If you are not using the WebLogic built-in JDBC drivers to set up the Oracle RTD Database, add the appropriate JDBC drivers bundled with Oracle RTD (*RTD_HOME*\lib*.jar) to the WebLogic classpath.

5.7 Creating a JDBC Provider for the Oracle RTD Database

Use the WebLogic Server Administration Console to create a JDBC provider for the Oracle RTD Database. Before you begin, ensure that WebLogic is started.

Notes: **1.** For general information about configuring WebLogic JDBC data sources, see *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*.

2. If your WebLogic version is 11g 10.3.3+ and your Oracle RTD database is in an Oracle database server, see Section 5.7.1, "Creating an Oracle JDBC Provider for the Oracle RTD Database on WebLogic 11g 10.3.3+."

To create a JDBC provider for the Oracle RTD Database:

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. Navigate the path **Services** -> **Data Sources** -> **New** -> **Generic Data Source**.
3. On the JDBC Data Source Properties page, follow these steps:
   a. For **Name**, enter a descriptive data source name, such as `RTD_DS`.
   b. For **JNDI Name**, enter `SDDS`.
   c. For **Database Type**, select **Other**.
   d. Ensure that the **Database Driver** is set to **Other**, then click **Next**.
4. On the Transaction Options page, deselect **Supports Global Transactions**, then click **Next**.
5. On the Connection Properties page, follow these steps:

   a. For **Database Name**, enter the name of the Oracle RTD Database you created in Section 2.2.

   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 (such as `1433` for SQL Server, `1521` for Oracle Database, or `50000` for DB2).

   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**.

6. On the Test Database Connection page, for **Driver Class Name**, enter the full package name of the JDBC driver class used to create the physical database connections in the connection pool (note that this driver class must be in the classpath of any server to which it is deployed):

   - **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`
   - **Oracle Database:** `oracle.jdbc.pool.OracleDataSource`
   - **DB2:** `com.ibm.db2.jcc.DB2Driver`

7. For **URL**, enter the URL of the database to which you want to connect. The format of the URL varies by database type:

   - **SQL Server:** `jdbc:sqlserver://`*`db_host`*`:`*`db_port`*

   > **Note:** If the database is on a SQL Server named instance, specify the *`db_host`* parameter using the format *`host_name`*`\`*`instance_name`*.

   - **Oracle Database:** `jdbc:oracle:thin:@`*`db_host`*`:`*`db_port`*`:`*`sid`*
   - **DB2:** `jdbc:db2://`*`db_host`*`:`*`db_port`*`/`*`db_name`*

8. In the Properties field, enter properties and their values required by the JDBC driver. The properties you need to provide vary by database type:

   - For **SQL Server**, enter the properties `user=`*`db_user_name`* and `DatabaseName=`*`db_name`* on separate lines
   - For **Oracle Database**, enter the property `user=`*`db_user_name`*
   - For **DB2**, enter the property `user=`*`db_user_name`*

9. Scroll to the bottom of the page and enter `SDAPPS` for **Test Table Name**.

10. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**.

11. Select the server where you want the data source to be made available (for example, **RTD_Server**). You must perform this step before completing the data source configuration.

12. Click **Finish**.

5.7.1 Creating an Oracle JDBC Provider for the Oracle RTD Database on WebLogic 11g 10.3.3+

To create a JDBC provider when your Oracle RTD database is in an Oracle database server, and Oracle RTD is deployed on WebLogic 11g 10.3.3+, perform the following steps:

1. Log into the WebLogic Server Administration Console with the administrator user name and password.
2. Navigate the path **Services** -> **Data Sources** -> **New** -> **Generic Data Source**.
3. On the JDBC Data Source Properties page, follow these steps:
   a. For **Name**, enter a descriptive data source name, such as `RTD_DS`.
   b. For **JNDI Name**, enter `SDDS`.
   c. For **Database Type**, select **Oracle**.
   d. For the **Database Driver,** select **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later**, then click **Next**.
4. On the Transaction Options page, deselect **Supports Global Transactions**, then click **Next**.
5. On the Connection Properties page, follow these steps:
   a. For **Database Name**, enter the name of the Oracle RTD Database you created in Section 2.2.
   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**.
6. On the Test Database Connection page, leave all the settings already filled, except enter `SDAPPS` for **Test Table Name**.
7. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**.
8. Select the server where you want the data source to be made available (for example, **RTD_Server**). You must perform this step before completing the data source configuration.
9. Click **Finish**, then click **Activate Changes**.

5.8 Creating Oracle RTD Roles and Users

Note: For general information about roles and permissions in Oracle RTD, see Chapter 7, "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 5.8.1, "Creating User Groups for Standard Roles"
- Section 5.8.2, "Creating User Groups for Custom Roles"
- Section 5.8.3, "Specifying Subgroups"
- Section 5.8.4, "Creating Roles"
- Section 5.8.5, "Creating the Oracle RTD Administrator and Other Users"
- *Section 5.8.6, "No Deployment Descriptor Update Required"*

5.8.1 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

5.8.2 Creating User Groups for Custom Roles

Create enterprise-specific groups, if Inline Service developement 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

5.8.3 Specifying Subgroups

Select each of the 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.

5.8.4 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 5–1 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 5–1.
   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 5–1.
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.

5.8.5 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.

5.8.6 No Deployment Descriptor Update Required

WebLogic does not require the updating of Oracle RTD's deployment descriptors to reference any custom roles.

5.9 Deploying Oracle RTD to the Managed Server

To deploy Oracle RTD to the managed server, 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 either Section 2.1.1, "Installing Oracle RTD on a Single Windows Computer" or Section 2.1.2, "Installing Real-Time Decision Server on UNIX."

   For example, *`RTD_HOME`*`\package\expanded`.

4. Set **Install this deployment as an application**.
5. Set the Name as OracleRTD.
6. In the Security area, select **Custom Roles**.

> **Note:** Use roles that are defined in the Administration Console; use policies that are defined in the deployment descriptor.

7. In the Source accessibility area, select **Copy this application onto every target for me** under **Recommended selection**.
8. Select the managed server that you created in Section 5.3, "Creating a Managed Server" as the target on which to deploy.

   After deployment, Oracle RTD is started automatically.

   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/log/server.log`.

5.10 Assigning Permissions to Custom Roles

As described in Section 7.4, "Assigning Permissions," assign Cluster permissions, Inline Service permissions, and Decision Center Perspective permissions to any custom roles.

5.11 Starting Oracle Real-Time Decisions

Restart Oracle RTD by stopping and then restarting the WebLogic server or cluster that Oracle RTD is deployed into.

5.12 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."

5.13 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 2.6 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 2.6.

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 2.6.
   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 2.6.
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 2.6.

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 Center and Load Generator. 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.

5.13.1 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 the following URL:

http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html

5.14 Setting Up JConsole Security for WebLogic

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. For the Arguments field, add the following, all on one line:

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

3. Save.

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

   > **Note:** If WebLogic is running on HP-UX or an AIX-based system, go to *`JDK_HOME`*`/jre/lib/management` instead, where *`JDK_HOME`* is the installation directory of the JDK being used by the WebLogic application server. See Section 1.3.4 for information about which JDK to use for WebLogic on those platforms.

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://java.sun.com/j2se/1.5.0/docs/guide/management/security-windows.html`

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

   For more information on JConsole security, go to:

`http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html#PasswordAccess`

> **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 15.1, "Accessing JConsole" for more information.

5.15 Changing the Oracle Real-Time Decisions Port Number in WebLogic

To change the Oracle RTD application port number in WebLogic, perform the following steps:

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, expand **Environment** and choose **Servers**.
3. Click the name of the managed server where Oracle RTD is running, for example, **RTD_Server**.
4. Click the **Configuration** tab, then the **General** subtab.
5. For **Listen Port**, enter the new port number you want to use for Oracle RTD.
6. Select **Listen Port Enabled**.
7. Click **Save**.
8. Restart the managed server on which Oracle RTD is running.

6

Configuring JBoss for Oracle Real-Time Decisions

Oracle RTD is supported on both UNIX and Windows platforms for the JBoss application server. The following sections explain how to install the Real-Time Decision Server on JBoss 4.3 EAR.

> **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.

> **Terminology:** In this chapter:
>
> - The directory into which you unzipped the JBoss 4.3 EAP archive file is referred to as *JBOSS_HOME*, for example, `C:\Program Files\EnterprisePlatform-4.3.0.GA_CP03`, `C:\jboss-eap-4.3`.
> - The directory into which you installed Oracle RTD is referred to as *RTD_HOME*, for example, `C:\OracleBI\RTD`.

This chapter contains the following topics:

- Section 6.1, "Configuring Server Properties"
- Section 6.2, "Creating a JDBC Provider for the Oracle RTD Database"
- Section 6.3, "Creating Oracle RTD Roles and Users."
- Section 6.4, "Installing and Starting the Oracle Real-Time Decisions Application on JBoss"
- Section 6.5, "Assigning Permissions to Custom Roles."
- Section 6.6, "Uninstalling the Oracle Real-Time Decisions Application from JBoss"
- Section 6.7, "Configuring SSL for Real-Time Decision Server (Recommended)"
- Section 6.8, "Setting Up JConsole for JBoss"

6.1 Configuring Server Properties

> **Note:** Make sure that you have performed the instructions in Section 1.4.4, "Installing JBoss for Oracle RTD."

This section consists of the following topics:

- Section 6.1.1, "Adding Oracle RTD Logging to the JBoss Server"
- Section 6.1.2, "Copying Oracle RTD Log 4J Jar File to JBoss"
- Section 6.1.3, "Copying New Version of Eclipse JDT Compiler to the JBoss Web Container Directory"
- Section 6.1.4, "Modifying the JBoss Start Script"

6.1.1 Adding Oracle RTD Logging to the JBoss Server

To add Oracle RTD logging to the JBoss Server, perform the following steps:

1. Open the file *`JBOSS_HOME`* `/jboss-as/server/default/conf/jboss-log4j.xml`.
2. Locate:

   ```
   <!-- ============================== -->
   <!-- Append messages to the console -->
   <!-- ============================== -->
   ```

   and add the following at the end of the above section:

   ```
   <!-- ============================================== -->
   <!-- START: Oracle RTD Appender Message Definitions -->
   <!-- ============================================== -->

   <appender name="SIGMA" class="com.sigmadynamics.util.SDRollingZipFileAppender">
      <param name="Threshold" value="DEBUG"/>
      <param name="File" value="${rtd.log.file}"/>
      <param name="Append" value="true"/>
      <param name="MaxFileSize" value="20000KB"/>
      <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{ISO8601} %-5p [%c{1}] %m%n"/>
      </layout>
      <filter class="org.jboss.logging.filter.TCLFilter">
         <param name="AcceptOnMatch" value="true"/>
         <param name="DeployURL" value="RTD.ear"/>
      </filter>
      <filter class="org.apache.log4j.varia.DenyAllFilter"/>
   </appender>

   <appender name="ENG_TRACE_FILE"
   class="com.sigmadynamics.util.SDRollingZipFileAppender">
      <param name="Threshold" value="TRACE"/>
      <param name="File" value="${rtd.log.file}.trace"/>
      <param name="Append" value="true"/>
      <param name="MaxFileSize" value="20000KB"/>
      <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern" value="%d{{HH:mm:ss,SSS}} [%t] %m%n"/>
        <!--
        <param name="ConversionPattern" value="%d{{HH:mm:ss,SSS}} [%t] %-5p
   ```

```
[%c{1}] %m%n"/>
     -->
   </layout>
</appender>

<!-- ============================================ -->
<!-- END: Oracle RTD Appender Message Definitions -->
<!-- ============================================ -->
```

3. Locate:

```
<!-- ====================== -->
<!-- More Appender examples -->
<!-- ====================== -->
```

and add the following at the end of the above section:

```
<!-- ============================================ -->
<!-- START: Oracle  Limit Category Definitions -->
<!-- ============================================ -->

<category name="ENG_TRACE" additivity="false">
  <priority value="DEBUG"/>
  <appender-ref ref="ENG_TRACE_FILE"/>
</category>

<category name="com.sigmadynamics.server.SDDistributedHashtable">
  <priority value="ERROR"/>
</category>

<category name="com.sigmadynamics.client">
  <priority value="ERROR"/>
</category>

<category name="com.sigmadynamics">
  <priority value="DEBUG"/>
</category>

<category name="org.jboss.messaging.core.impl.JDBCSupport">
  <priority value="INFO"/>
</category>

<category name="org.jboss.jms.server.plugin.JDBCJMSUserManager">
  <priority value="INFO"/>
</category>

<category name="org.jgroups.protocols">
  <priority value="ERROR"/>
</category>

<category name="sigmadynamics.application">
  <priority value="DEBUG"/>
</category>

<category name="request.trace.sigmadynamics.application">
  <priority value="TRACE#com.sigmadynamics.util.XLevel"/>
</category>

<category name="request.log.sigmadynamics.application">
  <priority value="DEBUG"/>
</category>
```

```
<!-- ========================================== -->
<!-- END: Oracle RTD Limit Category Definitions -->
<!-- ========================================== -->
```

4. Locate:

   ```
   <!-- ======================== -->
   <!-- Set up the Root category -->
   <!-- ======================== -->
   ```

   and add the following line to `<root>`:

   ```
   <appender-ref ref="SIGMA"/>
   ```

5. Save the file.

6.1.2 Copying Oracle RTD Log 4J Jar File to JBoss

Perform the following step, which adds Oracle RTD classes for displaying Oracle RTD server logs:

- Copy *RTD_HOME*/package/jboss/rtdlog4j.jar to *JBOSS_HOME*/jboss-as/server/default/lib/.

6.1.3 Copying New Version of Eclipse JDT Compiler to the JBoss Web Container Directory

Perform the following step:

- Copy *RTD_HOME*/package/jboss/ecj-3.4.jar to *JBOSS_HOME*/jboss-as/server/default/deploy/jboss-web.deployer/.

> **Note:** This overrides the older version of the Eclipse JDT compiler jasper-jdt.jar.
>
> See JBoss Bug #JBPAPP-869: https://jira.jboss.org/jira/browse/JBPAPP-869.

6.1.4 Modifying the JBoss Start Script

The JBoss start scripts for Windows and Unix are different. Follow the steps in the appropriate section:

- Section 6.1.4.1, "Modifying the Windows JBoss Start Script"
- Section 6.1.4.2, "Modifying the Unix JBoss Start Script"

6.1.4.1 Modifying the Windows JBoss Start Script

To modify the Windows JBoss start script, perform the following steps:

1. Open *JBOSS_HOME*/jboss-as/bin/run.bat.
2. Near the top of the file, add the following (after replacing *RTD_HOME_TO_REPLACE* with your actual value, for example, C:\OracleBI\RTD):

   ```
   set RTD_HOME=RTD_HOME_TO_REPLACE

   set JAVA_OPTS= -Dorg.jboss.net.protocol.file.decodeFilePaths=true
   set JAVA_OPTS=%JAVA_OPTS% -Dnologging=true -DSDLoggingPriority=DEBUG
   ```

```
set JAVA_OPTS=%JAVA_OPTS% -Djboss.partition.udpGroup=228.1.2.10
set JAVA_OPTS=%JAVA_OPTS% -Dhibernate.connection.release_mode=auto
set JAVA_OPTS=%JAVA_OPTS% -Drtd.install.dir=%RTD_HOME%
set JAVA_OPTS=%JAVA_OPTS% -Drtd.home.dir=%RTD_HOME%
set JAVA_OPTS=%JAVA_OPTS% -DSDLoggingFileName=%RTD_HOME%/log/server.log
set JAVA_OPTS=%JAVA_OPTS% -DDSPerfCounterLogFile=%RTD_HOME%/log/ds_perf.cvs
set JAVA_OPTS=%JAVA_OPTS% -Drtd.log.file=%RTD_HOME%/log/server.log
set JAVA_OPTS=%JAVA_OPTS% -Djava.net.preferIPv4Stack=true
set JAVA_OPTS=%JAVA_OPTS% -Xmx512m -XX:MaxPermSize=128m

set JBOSS_CLASSPATH=%RTD_HOME%/lib/jdbc/bijdbc14.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/sqljdbc.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/db2jcc_license_cu.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/db2jcc.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/ojdbc14.jar
```

3. Save the file.

> **Note:** To start the JBoss application server on Windows, perform the following:
>
> 1. Open a command window (cmd) and 'cd' to: *JBOSS_HOME*/jboss-as/bin/.
> 2. Start JBoss by entering:
>
>   run.bat --configuration=default --host=<*host or ip for JBoss to listen on*>
>
>   For example:
>
>   run.bat --configuration=default --host=0.0.0.0
>
> To stop the JBoss application server on Windows, perform the following:
>
> 1. Open a command window (cmd) and 'cd' to: *JBOSS_HOME*/jboss-as/bin/.
> 2. Stop JBoss by entering:
>
>   shutdown.bat --server=localhost:1099 -S -u admin -p admin

6.1.4.2 Modifying the Unix JBoss Start Script

To modify the Unix JBoss start script, perform the following steps:

1. Open *JBOSS_HOME*/jboss-as/bin/run.sh.
2. Near the top of the file, add the following (after replacing *RTD_HOME_TO_REPLACE* and *JAVAHOME_TO_REPLACE* with actual values, such as /home/rtd and /home/java):

```
JAVA_HOME="JAVAHOME_TO_REPLACE"
export JAVA_HOME

RTD_HOME="RTD_HOME_TO_REPLACE"

JAVA_OPTS="$JAVA_OPTS -Dorg.jboss.net.protocol.file.decodeFilePaths=true"
JAVA_OPTS="$JAVA_OPTS -Dnologging=true -DSDLoggingPriority=DEBUG"
JAVA_OPTS="$JAVA_OPTS -Djboss.partition.udpGroup=228.1.2.10"
JAVA_OPTS="$JAVA_OPTS -Dhibernate.connection.release_mode=auto"
JAVA_OPTS="$JAVA_OPTS -Drtd.install.dir=$RTD_HOME"
JAVA_OPTS="$JAVA_OPTS -Drtd.home.dir=$RTD_HOME"
```

```
JAVA_OPTS="$JAVA_OPTS -DSDLoggingFileName=$RTD_HOME/log/server.log"
JAVA_OPTS="$JAVA_OPTS -DDSPerfCounterLogFile=$RTD_HOME/log/ds_perf.cvs"
JAVA_OPTS="$JAVA_OPTS -Drtd.log.file=$RTD_HOME/log/server.log"
JAVA_OPTS="$JAVA_OPTS -Drtd.log.file=$RTD_HOME/log/server.log"
JAVA_OPTS="$JAVA_OPTS -Djava.net.preferIPv4Stack=true
JAVA_OPTS="$JAVA_OPTS -Xmx512m -XX:MaxPermSize=128m -server

JBOSS_CLASSPATH="$RTD_HOME/lib/jdbc/bijdbc14.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/sqljdbc.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/db2jcc_license_cu.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/db2jcc.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/ojdbc14.jar"
```

Tips: **1.** If you get the error OutOfMemoryError, set the `-Xmx` argument to `-Xmx1024M`.

2. If you get the error OutOfMemoryError:PermGen, set the `-XX:MaxPermSize` argument to `-XX:MaxPermSize=256M`.

3. Save the file.

Note: To start the JBoss application server on Unix, perform the following:

1. Open a command window (cmd) and 'cd' to: *JBOSS_HOME*/jboss-as/bin/.
2. Start JBoss by entering:

   `run.sh --configuration=default --host=<`*host or ip for JBoss to listen on*`>`

   For example:

   `run.sh --configuration=default --host=0.0.0.0`

To stop the JBoss application server on Unix, perform the following:

1. Open a command window (cmd) and 'cd' to: *JBOSS_HOME*/jboss-as/bin/.
2. Stop JBoss by entering:

   `shutdown.sh --server=localhost:1099 -S -u admin -p admin`

6.2 Creating a JDBC Provider for the Oracle RTD Database

To create a JDBC provider for the Oracle RTD database, set up the Oracle RTD SDDS data source and to copy it to the JBoss deploy directory, as follows:

1. In *RTD_HOME*/package/jboss, choose your supported database and open the file appropriate to that database, as follows:
   - For DB2: `rtd-db2-ds.xml`
   - For Oracle: `rtd-oracle-ds.xml`
   - For SQL Server: `rtd-sqlserver-ds.xml`
2. Replace the following:
   - `${DB_SERVER}` with your database server

     For example: `localhost`

Note: If the database is on a SQL Server named instance, specify the name of your database server using the format *host_name\instance_name*.

- `${DB_PORT}` with your database port

  For example: `50000` for db2, `1521` for oracle, `1433` for sqlserver
- `${DB_NAME}` with your database name

  For example: `RTD` for db2 and sqlserver, `orcl` for oracle
- `${DB_USER}` with your database user name

  For example: `jsmith`
- `${DB_PASSWORD}` with your database user password

3. Save the file and copy it to *JBOSS_HOME*/`jboss-as/server/default/deploy/`.

6.3 Creating Oracle RTD Roles and Users

Note: For general information about roles and permissions in Oracle RTD, see Chapter 7, "Configuring Security for Oracle Real-Time Decisions."

In JBoss, the default security provider uses two files to define users and roles, respectively. These files should be edited, or created if they do not already exist, to include the users that are to access Oracle RTD, and their enterprise roles.

6.3.1 Creating Oracle RTD Administrators and Other Users

Assuming JBoss is installed in the directory signified by *JBOSS_HOME*, then edit this file to include the names and passwords (in clear-text) of users that will access Oracle RTD:

- *JBOSS_HOME*/`jboss-as/server/default/conf/users.properties`

Note: Because the default security provider that comes with JBoss stores user names and passwords in an unencrypted form, Oracle highly recommends some other identity store be used with JBoss.

The following shows an example of the content of this file, to demonstrate the syntax. The user name and password appear on the same line, separated by an '=' character.

```
# A users.properties file for use with the UsersRolesLoginModule
sdsu=welcome1
don=donSecret1
jane=janeSecret1
```

The users are defined in the following table:

Table 6–1 Users

User Name	Password	Description
sdsu	welcome1	An Oracle RTD administrator
don	donSecret1	An Oracle RTD developer
jane	janeSecret1	An Oracle RTD business

6.3.2 Creating Standard Roles

You create Roles in the default security provider of JBoss by editing this file:

- *JBOSS_HOME*/jboss-as/server/default/conf/roles.properties

Note: Create the file if it does not already exist.

In the file, each user and the corresponding assigned roles should appear on a single line.

```
# A roles.properties file for use with the UsersRolesLoginModule
#
sdsu=RTDUsers,RTDAdministrators
don=RTDUsers,RTDStudioDeployers,RTDDecsionCenterUsers
jane=RTDUsers,RTDDecisionCenterEditors
```

The following are the roles defined as in the previous list, all of which are standard roles:

- RTDUsers
- RTDAdministrators
- RTDDecisionCenterEditors
- RTDDecisionCenterUsers
- RTDStudioDeployers

The following are the Oracle RTD standard roles not referenced by the example, and hence not created:

- RTDStudioDownloaders
- RTDBatchAdministrators
- RTDChoiceEditors

6.3.3 Creating Custom Roles

Define custom roles in the same manner as standard roles, by editing the file `roles.properties`.

Note: Create the file if it does not already exist.

6.3.4 Updating Oracle RTD’s Deployment Descriptors

Two of Oracle RTD's deployment descriptor files must be updated to reference any custom roles. These files are embedded within `ui.war` and `soap.war`, both of which are embedded within `RTD.ear`.

> **Note:** This section describes the changes that need to be made to two `web.xml` files, but does not describe precisely how to find and update those files. The process is similar to the process of updating the `web.xml` files to contain custom database resource references, as described in Section 8.4.2, "Adding the New Data Source to Oracle RTD."

Update ui.war

The file represented by this path must be updated to reference any custom roles:

- `rtd.ear/`**`ui.war`**`/web-inf/web.xml`

There are three places within the file `web.xml` that need to be updated to reference custom roles. Look for the three occurrences of the string, "RTD-Custom-Roles", inside comments.

The first two places are for inserting security-role-ref elements, as in the following:

```
 <!-- RTD-Custom-Roles. Insert custom RTD roles here, as might be needed for
         ILS-specific access. For example, RTD_ILS2Developers could be a role for
         deploying the Inline Service named ILS2, and RTD_ILS2Users for viewing or
         editing ILS2 in Decision Center.
 -->
      <security-role-ref>
          <role-name>RTD_ILS2Developers</role-name>
          <role-link>RTD_ILS2Developers</role-link>
      </security-role-ref>
      <security-role-ref>
          <role-name>RTD_ILS2Users</role-name>
          <role-link>RTD_ILS2Users</role-link>
      </security-role-ref>
```

The third place for custom roles is for inserting security-role elements, as in the following:

```
<!-- RTD-Custom-Roles. Insert here security-role elements for any custom RTD
roles,
         as might be needed for ILS-specific access. For example, RTD_
ILS2Developers
         could be a role for deploying the Inline Service named ILS2, and RTD_
ILS2Users
         for viewing or editing ILS2 in Decision Center.
-->
      <security-role>
          <role-name>RTD_ILS2Developers</role-name>
      </security-role>
      <security-role>
          <role-name>RTD_ILS2Users</role-name>
      </security-role>
```

Update soap.war

The file represented by this path must be updated to reference any custom roles:

- `rtd.ear/`**`soap.war`**`/web-inf/web.xml`

There are two places within the file `web.xml` that must be updated to reference custom roles. Look for the two occurrences of the string, "RTD-Custom-Roles", inside comments.

The first place is for inserting security-role-ref elements, as in the following:

```
 <!-- RTD-Custom-Roles. Insert custom RTD roles here, as might be needed for
        ILS-specific access. For example, RTD_ILS2Developers could be a role for
        deploying the Inline Service named ILS2, and RTD_ILS2Users for viewing or
        editing ILS2 in Decision Center.
-->
      <security-role-ref>
          <role-name>RTD_ILS2Developers</role-name>
          <role-link>RTD_ILS2Developers</role-link>
      </security-role-ref>
      <security-role-ref>
          <role-name>RTD_ILS2Users</role-name>
          <role-link>RTD_ILS2Users</role-link>
      </security-role-ref>
```

The second place for custom roles is for inserting security-role elements, as in the following:

```
<!-- RTD-Custom-Roles. Insert here security-role elements for any custom RTD
roles,
         as might be needed for ILS-specific access. For example, RTD_
ILS2Developers
         could be a role for deploying the Inline Service named ILS2, and RTD_
ILS2Users
         for viewing or editing ILS2 in Decision Center.
-->
      <security-role>
          <role-name>RTD_ILS2Developers</role-name>
      </security-role>
      <security-role>
          <role-name>RTD_ILS2Users</role-name>
      </security-role>
```

6.4 Installing and Starting the Oracle Real-Time Decisions Application on JBoss

To install the Oracle RTD on JBoss, copy the Oracle `RTD.ear` file to the JBoss deploy directory, as follows:

- Copy *RTD_HOME*`/package/RTD.ear` to *JBOSS_HOME*`/jboss-as/server/default/deploy/`.

After you start JBoss, Oracle RTD is automatically started.

6.5 Assigning Permissions to Custom Roles

As described in Section 7.4, "Assigning Permissions," assign Cluster permissions, Inline Service permissions, and Decision Center Perspective permissions to any custom roles.

6.6 Uninstalling the Oracle Real-Time Decisions Application from JBoss

To uninstall Oracle RTD from JBoss, you must remove `RTD.ear` file from the JBoss deploy directory when JBoss has been stopped, as follows:

1. Stop JBoss.

2. Remove the `RTD.ear` file from the JBoss deploy directory, *`JBOSS_HOME`*`/jboss-as/server/default/deploy/`.
3. Start JBoss.

For information about starting and stopping JBoss on your operating system, see the Note - which refers to *both* start and stop scripts - at the end of the section appropriate to your installation:

- Section 6.1.4.1, "Modifying the Windows JBoss Start Script"
- Section 6.1.4.2, "Modifying the Unix JBoss Start Script"

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

The process of setting up SSL for all client connections to Real-Time Decision Server takes place automatically as you install JBoss for Oracle RTD. For details, see Section 1.4.4, "Installing JBoss for Oracle RTD."

> **Note:** If you want to use your own keystore and truststore, you do not need to complete the instructions in Section 2.6.

6.8 Setting Up JConsole for JBoss

To set up JConsole for JBoss, follow the instructions appropriate for your operating system:

- Section 6.8.1, "Setting Up JConsole for JBoss on Windows"
- Section 6.8.2, "Setting Up JConsole for JBoss on Unix"

6.8.1 Setting Up JConsole for JBoss on Windows

To set up JConsole for JBoss on Windows, perform the following steps:

1. Open *`JBOSS_HOME`*`/jboss-as/bin/run.bat`.
2. Near the top of the file, add the following:

```
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote=true
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=12345
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false
```

3. Save the file.

6.8.2 Setting Up JConsole for JBoss on Unix

To set up JConsole for JBoss on Unix, perform the following steps:

1. Open *`JBOSS_HOME`*`/jboss-as/bin/run.sh`.
2. Near the top of the file, add the following:

```
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote=true"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=12345"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.ssl=false"
```

3. Save the file.

7 Configuring Security for Oracle Real-Time Decisions

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:

7.1 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 7.1.1, "Cluster Permissions"
- Section 7.1.2, "Inline Service Permissions"
- Section 7.1.3, "Decision Center Perspective Permissions"

7.1.1 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 7–1 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

7.1.2 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 7–2 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

7.1.3 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 7–3 lists the default Decision Center perspectives.

Table 7–3 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.

7.2 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 7.2.1, "Default Cluster Permission Assignments"
- Section 7.2.2, "Default Inline Service Permissions"
- Section 7.2.3, "Default Decision Center Perspective Permissions"

7.2.1 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 7–4 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.

7.2.2 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 7–5 shows the default Inline Service permissions assigned to the standard RTD roles.

Table 7–5 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

7.2.3 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 7–6.

Table 7–6 Decision Center Perspective Permissions

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

7.3 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 7–7.

Table 7–7 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.

7.4 Assigning Permissions

Section 7.2, "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 7.4.1, "Granting Cluster Permissions"
- Section 7.4.2, "Granting Inline Service Permissions"
- Section 7.4.3, "Granting Decision Center Perspective Permissions"

7.4.1 Granting Cluster Permissions

Cluster permissions are granted through JMX, using the MBean whose path is OracleRTD > SDManagement > Security Manager. For more details, see Section 15.5, "Oracle Real-Time Decisions Security Management."

7.4.1.1 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

7.4.1.2 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.

7.4.1.3 Operation listDirectPermissions

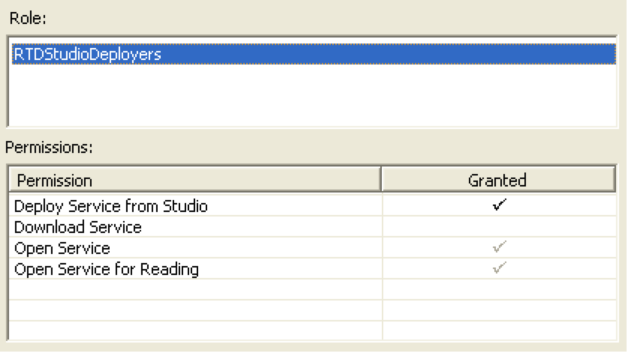
The **listDirectPermissions** operation shows all the cluster permissions that have been granted to a role, including permissions granted directly by the **assignPermission** operation, and any implied permissions.

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

7.4.2 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.

7.4.3 Granting Decision Center Perspective Permissions

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*.

7.5 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

7.6 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 7.6.1, "Migrating Oracle RTD Users and Groups to Enterprise Identity Store"
- Section 7.6.2, "Defining Standard Security Roles in JEE Server"
- Section 7.6.3, "Defining Custom Security Roles in JEE Server"
- Section 7.6.4, "Declaring Custom Role References"
- Section 7.6.5, "Assigning Inline Service Permissions to Roles"
- Section 7.6.6, "Assigning Decision Center Perspective Permissions to Roles"

7.6.1 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.

7.6.2 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.

7.6.3 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 7–8 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

7.6.4 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`

7.6.5 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.

7.6.6 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.

8

Configuring Data Access for Oracle Real-Time Decisions

JDBC data sources are used by Oracle RTD to access outside data. Data sources are application-server specific, and are used to identify new JDBC data sources that are to be used as suppliers in Inline Services. These data sources can be RDBMS databases, as well as ODBC identified data sources. For detailed information about supported versions for enterprise data sources, refer to *Oracle Real-Time Decisions System Requirements and Supported Platforms* for Version 3.0.

This chapter contains the following topics:

- Section 8.1, "Creating Additional JDBC Data Sources in OC4J"
- Section 8.2, "Creating Additional JDBC Data Sources in WebSphere"
- Section 8.3, "Creating Additional JDBC Data Sources in WebLogic"
- Section 8.4, "Creating Additional JDBC Data Sources in JBoss"
- Section 8.5, "Testing a New Enterprise Data Source"

8.1 Creating Additional JDBC Data Sources in OC4J

If you are running Real-Time Decision Server on OC4J, follow the steps in this section to configure JDBC data sources so that your Inline Services can access outside data.

This section contains the following topics:

- Section 8.1.1, "Providing Access to JDBC Jar Files for Your Data Source"
- Section 8.1.2, "Creating a Connection Pool and Data Source in OC4J"
- Section 8.1.3, "Adding the New Data Source to Oracle RTD"

8.1.1 Providing Access to JDBC Jar Files for Your Data Source

Follow the steps in this section to give OC4J access to the JDBC jar files for your data source. The jar file names vary, depending on the type of database you are adding as a data source.

To provide access to JDBC jar files:

1. If you are using standalone OC4J, copy the appropriate jar files to `*OC4J_HOME*/j2ee/home/applib`. If you are using Oracle Application Server, copy the appropriate jar files to `*ORACLE_AS_HOME*/j2ee/*oc4j_instance*/applib`. Copy the jars appropriate for your data source:

- For **SQL Server**, copy the file `sqljdbc.jar` in the directory *RTD_HOME*/`lib/jdbc`.
- For **DB2**, copy the files `db2jcc.jar` and `db2jcc_license_cu.jar` in *RTD_HOME*/`lib/jdbc`.
- For **Teradata**, For Teradata, copy the files `tdgssconfig.jar` and `terajdbc4.jar`.

  You can download these files from the Teradata Web site at `http://www.teradata.com`. Make sure the jar files are from Teradata JDBC Driver for Teradata 12.0. This JDBC driver supports both Teradata V2R6 and 12.0.
- For **Oracle BI EE**, copy the file `bijdbc.jar`.

  You can find this file in your Oracle BI EE installation.

2. Restart OC4J.

You do not need to copy any jar files if you are adding an Oracle Database or Oracle's Siebel Analytics Server data source.

8.1.2 Creating a Connection Pool and Data Source in OC4J

You can use the Application Server Control to create a connection pool and data source in OC4J. Before you begin, ensure that OC4J is started.

To create a connection pool and data source in OC4J:

1. Log in to the Application Server Control as the `oc4jadmin` user. You can access the Application Server Control at `http://`*`oc4j_host`*`:`*`port`*`/em`. For the standalone version of OC4J, the port number is typically `8888`.
2. Click the **Administration** tab. Then, under the Services heading, click the **JDBC Resources** task. Under the Connection Pools heading, click **Create** to create a new connection pool for your enterprise database.

   If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Administration** tab.
3. On the Create Connection Pool - Application page, ensure the following:

   - For SQL Server and DB2 connection pools, **default** is selected for **Application**

     For Oracle connection pools, **OracleRTD** is selected for **Application**
   - **New Connection Pool** is selected for **Connection Pool Type**.

   Then, click **Continue**.

   > **Note:** If the enterprise data source you want to add is the same type of database as your Oracle RTD Database, you may want to select **New Connection Pool from Existing Connection Pool** for **Connection Pool Type**. Then, select **"RTDConnectionPool"** from the **Existing Connection Pool** list.
4. On the Create Connection Pool page, enter *`db_name`*`_Pool` for **Name** (for example, `mydatabase_Pool`).
5. For Connection Factory Class, enter one of the following:

   - **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`

- **Oracle Database:** `oracle.jdbc.pool.OracleDataSource`
- **DB2:** `com.ibm.db2.jcc.DB2Driver`
- **Teradata:** `com.teradata.jdbc.TeraDriver`
- **Oracle BI EE:** `oracle.bi.jdbc.AnaJdbcDriver`
- **Siebel Analytics Server:** `sun.jdbc.odbc.JdbcOdbcDriver`

6. For **JDBC URL**, enter one of the following:

   - **SQL Server:**

     ```
     jdbc:sqlserver://db_host:db_port;databaseName=db_name
     ```

     Note: If the data source is on a SQL Server named instance, specify the *db_host* parameter using the format *host_name\instance_name*.

   - **Oracle Database:**

     ```
     jdbc:oracle:thin:@db_host:db_port:sid
     ```

   - **DB2:**

     ```
     jdbc:db2://db_host:db_port/db_name
     ```

   - **Teradata:**

     ```
     jdbc:teradata://server_name:port/db_name/param1,param2,...
     ```

     Note: If *db_name* is missing, the current login user's default database is used. For example, with default database **RTD30**, `jdbc:teradata://64.181.232.117//TMODE=ANSI,CHARSET=ASCII` executes as `jdbc:teradata://64.181.232.117/`**`RTD30`**`/TMODE=ANSI,CHARSET=ASCII`

   - **Oracle BI EE:**

     ```
     jdbc:oraclebi://server_name:9703/user=bi_user_name;password=bi_password;
     catalog=catalog_name;
     ```

     The catalog name is optional.

   - **Siebel Analytics Server:**

     ```
     jdbc:odbc:SAS_DSN_name
     ```

     Note: To add a Siebel Analytics Server data source, you must first create and configure a Siebel Analytics ODBC data source name (*SAS_DSN_name*). Refer to *Siebel Analytics Server Administration Guide* for complete details. The Siebel Analytics Server data source name cannot contain spaces.

7. Under the Credentials heading, for **Username**, provide a database user name with run-time privileges for the enterprise instance. Then, provide the corresponding

password. See *Oracle Fusion Middleware Security Guide* for information about whether to provide a cleartext password or an indirect password.

If you are adding an Oracle BI EE or Siebel Analytics Server data source, provide credentials for Oracle BI EE or Siebel Analytics Server rather than a database user name.

8. Click **Finish**.
9. On the JDBC Resources page, under the Data Sources heading, click **Create** to define a new data source.
10. On the Create Data Source - Application & Type page, ensure the following:
    - For SQL Server and DB2 data sources, **default** is selected for **Application**

      For Oracle data sources, **OracleRTD** is selected for **Application**
    - **Managed Data Source** is selected for **Data Source Type**

    Then, click **Continue**.

    > **Note:** If the enterprise data source you want to add is the same type of database as your Oracle RTD Database, you may want to select **New Data Source from Existing Data Source** for **Data Source Type**. Then, select **SDDS** from the **Existing Data Source** list.

11. On the Create Data Source - Managed Data Source page, follow these steps:
    a. For **Name**, enter a name for the data source (for example, the name of your database).

    b. For **JNDI Location**, enter a *`jndi_name`* of your choice. The *`jndi_name`* you provide will be used by Oracle RTD to look up the data source.

    c. For **Connection Pool**, select ***db_name*_Pool**.

    d. Click **Finish**.
12. On the JDBC Resources page, in the Data Sources table, click **Test Connection** for the enterprise data source you just added.

    > **Note:** Do not test the connection if you are adding an Oracle BI EE or Siebel Analytics Server data source.

    Follow these steps to test the data source:

    a. If you are adding an Oracle Database data source, keep the default settings and click **Test**.

    b. If you are adding a SQL Server or DB2 data source, change the **SQL Statement** to `select * from` *`your_table_name`*, where *`your_table_name`* is the name of an existing table in the database, then click **Test**.

    c. If you are adding a Teradata data source, for test connection purposes, it is better to test a small table. Change the **SQL Statement** to either `select count(*) from` *`your_table_name`* or `LOCK TABLE` *`your_table_name`* `FOR READ SELECT * FROM` *`your_table_name`*. Then click **Test**.

    If the connection is not established successfully, restart OC4J and then test the data source again. If it still fails, ensure that your connection pool settings are correct.

8.1.3 Adding the New Data Source to Oracle RTD

After you create a connection pool and data source in OC4J, you must add the new data source to Oracle RTD. To do this, you must add a resource reference to four different XML files. Before you begin, ensure that OC4J is not running.

To add the new data source to Oracle RTD:

1. If Oracle RTD is running on standalone OC4J, go to the directory *OC4J_HOME*/j2ee/home/applications/OracleRTD. If Oracle RTD is running on Oracle Application Server, go to *ORACLE_AS_HOME*/j2ee/*oc4j_instance*/applications/OracleRTD.

   This directory is the location where the RTD.ear file was expanded when you deployed it as an application.

2. Locate the file ./rtis/web.xml and open it for editing. Scroll to the bottom of the file. Copy the section for the definition of the resource reference of SDDS_RTIS and paste it after the existing section. In the copied section, replace the string SDDS with the JNDI name (*jndi_name*) you entered in Step 11 of Section 8.1.2. For example:

   ```
   <resource-ref id="jndi_name_RTIS">
   <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>
   ```

   Then, save the changes and close the file.

3. Locate the file ./soap/web.xml and open it for editing. Scroll to the bottom of the file. Copy the section for the definition of the resource reference for SDDS_Axis and paste it after the existing section. In the copied section, replace the string SDDS with the JNDI name (*jndi_name*) you entered in Step 11 of Section 8.1.2. For example:

   ```
   <resource-ref id="jndi_name_Axis">
   <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>
   ```

   Then, save the changes and close the file.

4. Go to the directory *OC4J_HOME*/j2ee/home/application-deployments/OracleRTD. Make sure to go to the application-deployments directory, not the applications directory.

5. Locate the file ./soap/orion-web.xml and open it for editing. Scroll to the bottom of the file. Copy the line for the definition of the resource reference mapping of SDDS and paste it after the existing line. In the copied line, replace the string SDDS with the JNDI name (*jndi_name*) you entered in Step 11 of Section 8.1.2. For example:

   ```
   <resource-ref-mapping name="jndi_name" location="jndi_name"/>
   ```

   Then, save the changes and close the file.

6. Locate the file ./rtis/orion-web.xml and open it for editing. Scroll to the bottom of the file. Copy the line for the definition of the resource reference

mapping of SDDS and paste it after the existing line. In the copied line, replace the string SDDS with the JNDI name (*jndi_name*) you entered in Step 11 of Section 8.1.2. For example:

```
<resource-ref-mapping name="jndi_name" location="jndi_name"/>
```

Then, save the changes and close the file.

7. Start OC4J.

8.2 Creating Additional JDBC Data Sources in WebSphere

If you are running Real-Time Decision Server on WebSphere, follow the steps in this section to configure JDBC data sources so that your Inline Services can access outside data.

This section contains the following topics:

- Section 8.2.1, "Creating a Data Provider and Data Source in WebSphere"
- Section 8.2.2, "Adding the New Data Source to Oracle RTD"
- Section 8.2.3, "Updating Resource References in WebSphere"
- Section 8.2.4, "Adding Additional Data Sources"

8.2.1 Creating a Data Provider and Data Source in WebSphere

You can use the Integrated Solutions Console to create a data provider and data source in WebSphere. Before you begin, ensure that WebSphere is started.

To create a data provider and data source in WebSphere:

1. If the enterprise data source you want to add is a Teradata data source, you must first copy the files `terajdbc4.jar`, `tdgssjava.jar`, and `tdgssconfig.jar` to *RTD_HOME*`/lib/jdbc`. You can download these files from the Teradata Web site at `http://www.teradata.com`. Make sure the jar files are version 3.3.0.2 or later.
2. 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**.
3. In the tree on the left, expand **Resources**, then expand **JDBC** and choose **JDBC Providers**.
4. Ensure that the scope is set correctly (to **Node=*host_name*Node_*number***), then click **New**.
5. For **Database type**, select **User-defined**.
6. For Implementation class name, enter one of the following values:
   - **SQL Server:**

     ```
     com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource
     ```

   - **Oracle Database:**

     ```
     oracle.jdbc.pool.OracleConnectionPoolDataSource
     ```

   - **DB2:**

```
com.ibm.db2.jcc.DB2ConnectionPoolDataSource
```

- **Teradata:**

```
com.teradata.jdbc.TeraConnectionPoolDataSource
```

7. For **Name**, enter a name for this provider (such as the name of the database). You can also enter an optional description (recommended). Then, click **Next**.
8. For **Class path**, enter one of the following values:
   - **SQL Server:**

     ```
     RTD_HOME/lib/jdbc/sqljdbc.jar
     ```

   - **Oracle Database:**

     ```
     RTD_HOME/lib/jdbc/ojdbc14.jar
     ```

   - **DB2:**

     ```
     RTD_HOME/lib/jdbc/db2jcc.jar;RTD_HOME/lib/jdbc/db2jcc_license_cu.jar
     ```

   - **Teradata:**

     ```
     RTD_HOME/lib/jdbc/terajdbc4.jar;RTD_HOME/lib/jdbc/tdgssconfig.jar
     ```

   Make sure to replace `RTD_HOME` with the actual Oracle RTD installation path.
9. Click **Next**, then click **Finish**, then click **Save**.
10. Create a J2C authentication entry for the data source, as follows:
    a. In the tree on the left, expand **Security** and choose **Secure administration, applications, and infrastructure**.
    b. Under the Authentication heading, expand **Java Authentication and Authorization Service** and choose **J2C authentication data**. Then, click **New**.
    c. For **Alias**, enter a descriptive name for the authentication entry (such as *db_name_auth*).
    d. For **User ID** and **Password**, enter the user name and password for the enterprise database.
    e. Click **OK**, then click **Save**.
11. In the tree on the left, expand **Resources**, then expand **JDBC** and choose **JDBC Providers**. Then, click the name of the JDBC provider you added for this database.
12. Under the Additional Properties heading, click **Data sources**.
13. Create a new data source, as follows:
    a. Click **New**.
    b. For **Data source name**, provide a descriptive name for the data source (for example, *db_name*`_DS`.
    c. For **JNDI name**, enter the same value you provided for **Data source name**. This value will appear in Decision Studio when you perform an Import in a data source object.
    d. For **Component-managed authentication alias and XA authentication**, select the J2C authentication alias you created in Step 10.
    e. Click **Next**, then click **Finish**, then click **Save**.

14. Click the name of the data source you just added.

15. If you are adding a SQL Server data source, follow these steps to set properties for the data source:

    a. Under the Additional Properties heading, click **Custom properties**.

    b. Click the **Select All** icon, then click **Delete** to delete any existing custom properties.

    c. Click **New**, enter `databaseName` for **Name**, then enter the name of your database for **Value**. Then, click **OK**.

    d. Click **New**, enter `portNumber` for **Name**, then enter the port number of your database (typically `1433`) for **Value**. Then, click **OK**.

    e. Click **New**, enter `serverName` for **Name**, then enter the name of your database server for **Value**.

    > **Note:** If the data source is on a SQL Server named instance, specify the name of your database server using the format *`host_name\instance_name`*.

    f. Click **OK**.

    g. Click **Save**.

16. If you are adding an Oracle Database data source, follow these steps to set properties for the data source:

    a. Scroll to the bottom of the page and locate the Oracle data source properties section.

    b. For **URL**, enter `jdbc:oracle:thin:@`*`db_host`*`:`*`db_port`*`:`*`sid`*.

       For example:

       ```
       jdbc:oracle:thin:@dbhost.company.com:1521:orcl
       ```

    c. Click **OK**, then click **Save**.

17. If you adding a DB2 data source, follow these steps to set properties for the data source:

    a. Scroll to the bottom of the page and locate the DB2 Universal data source properties section.

    b. For **Database name**, enter the name of your database.

    c. For **Driver type**, enter `4`.

    d. For **Server name**, enter the name of your database server.

    e. For **Port number**, enter the port number of your database.

    f. Click **OK**, then click **Save**.

18. If you are adding a Teradata data source, follow these steps to set properties for the data source:

    a. Under the Additional Properties heading, click **Custom properties**.

    b. Click **DSName**, enter the name of the Teradata database server host for **Value**, and click **OK**.

c. Click **databaseName**, enter the name of the Teradata database for **Value**, and click **OK**.

d. Click **Save**.

8.2.2 Adding the New Data Source to Oracle RTD

After you create a data provider and data source in WebSphere, you must add the new data source to Oracle RTD. To do this, you must add a resource reference to two different `web.xml` files, as follows:

1. Open the following file for editing:

   ```
   WEBSPHERE_HOME/AppServer/profiles/profile_name/config/cells/server_name/
   applications/OracleRTD.ear/deployments/OracleRTD/rtis.war/WEB-INF/web.xml
   ```

2. 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 `_RTIS`, 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 you provided in Step 13 of Section 8.2.1.

   The following example shows a new `<resource-ref>` entry for a Teradata data source:

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

3. After you have completed the new entry, save and close the file.

4. Open the following file for editing:

   ```
   WEBSPHERE_HOME/AppServer/profiles/profile_name/config/cells/server_name/
   applications/OracleRTD.ear/deployments/OracleRTD/soap.war/WEB-INF/web.xml
   ```

5. 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 `_Axis`, 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 you provided in Step 13 of Section 8.2.1.

   The following example shows a new `<resource-ref>` entry for a Teradata data source:

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

6. After you have completed the new entry, save and close the file.

8.2.3 Updating Resource References in WebSphere

After you add resource references to the `web.xml` files, you must configure the resource references in WebSphere using the Integrated Solutions Console. Before you begin, ensure that WebSphere is started.

To update resource references in WebSphere:

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 **Applications**, then choose **Enterprise Applications**. Then, click **Oracle RTD**.
3. Under the References heading, click **Resource references**.
4. Scroll down to the table that shows the resource references and locate the two entries that you added to the `web.xml` files.
5. For each entry, set the **Target Resource JNDI Name** to the JNDI name you provided in Step 13 of Section 8.2.1.
6. Select both entries, then scroll up to the Specify authentication method heading and select **Use default method (many-to-one mapping)**. Then, for Authentication data entry, select the J2C authentication entry you created in Step 10 of Section 8.2.1.
7. Click **Apply**.
8. Scroll to the bottom of the table and click **OK**, then click **Save**.

8.2.4 Adding Additional Data Sources

If you want to add another data source of the same type, start from Step 10 of Section 8.2.1. You do not need to create an additional JDBC provider if you have already created one for a particular database type.

If you want to add another data source of a different type, you must repeat all the steps.

8.3 Creating Additional JDBC Data Sources in WebLogic

If you are running Real-Time Decision Server on WebLogic, follow the steps in this section to configure JDBC data sources so that your Inline Services can access outside data.

This section contains the following topics:

- Section 8.3.1, "Setting the Path to JDBC Jar Files for Your Data Source"
- Section 8.3.2, "Creating a Data Source in WebLogic"
- Section 8.3.3, "Adding the New Data Source to Oracle RTD"

8.3.1 Setting the Path to JDBC Jar Files for Your Data Source

To set the path to JDBC jar files for your data source, you may need to add paths to the Classpath for your managed server, if your data source database is different from the Oracle RTD database, and you have not previously edited the Classpath.

1. If the enterprise data source you want to add is a Teradata data source, copy the files `terajdbc4.jar`, and `tdgssconfig.jar` to *`RTD_HOME`*`/lib/jdbc`. You can download these files from the Teradata Web site at `http://www.teradata.com`. Make sure the jar files are from Teradata JDBC Driver for Teradata 12.0. This JDBC driver supports both Teradata V2R6 and 12.0.

2. If the enterprise data source you want to add is an Oracle BI EE data source, copy the file `bijdbc.jar` to *`RTD_HOME`*`/lib/jdbc`. You can find this file in your Oracle BI EE installation.

3. Log in to the Administration Console for your Oracle RTD domain.

   `http://`*`weblogic_host`*`:`*`port`*`/console`.

4. Navigate the path:

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

5. Add the appropriate path or paths to **ClassPath**, replacing RTD_HOME with your Oracle RTD directory, and including Windows or Linux/Unix separators between entries. Do not include spaces between your entries.

   - For Oracle

     `RTD_HOME``\lib\jdbc\ojdbc14.jar`

   - For SQLServer

     `RTD_HOME``\lib\jdbc\sqljdbc.jar`

   - For DB2

     `RTD_HOME``\lib\jdbc\db2jcc.jar`

     `RTD_HOME``\lib\jdbc\db2jcc_license_cu.jar`

   - For Teradata

     `RTD_HOME``\lib\jdbc\terajdbc4.jar`

     `RTD_HOME``\lib\jdbc\tdgssconfig.jar`

   - For Oracle BI EE

     `RTD_HOME``\lib\jdbc\bijdbc.jar`

   Note: If it is not already specified in the **Classpath**, add the entry *`MW_HOME`*`\wlserver_10.3\server\lib\weblogic.jar`, after replacing *`MW_HOME`* with the directory where WebLogic is installed.

6. Save.

8.3.2 Creating a Data Source in WebLogic

You can use the WebLogic Server Administration Console to create a data source in WebLogic. Before you begin, ensure that WebLogic is started.

Notes: 1. For general information about configuring WebLogic JDBC data sources, see *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*.

2. If your WebLogic version is 11g 10.3.3+ and your Oracle RTD database is in an Oracle database server, see Section 8.3.2.1, "Creating an Oracle JDBC Provider for the Oracle RTD Database on WebLogic 11g 10.3.3+."

To create a data source in 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. Navigate the path **Services** -> **Data Sources** -> **New** -> **Generic Data Source**.
3. Click **New**. You may need to click **Lock & Edit** first to enable the **New** button.
4. On the JDBC Data Source Properties page, follow these steps:
   a. For **Name**, provide a descriptive name for the data source (for example, *`db_name`*`_DS`).
   b. For **JNDI Name**, enter the same value you provided for **Name**. This value will appear in Decision Studio when you perform an Import in a data source object.
   c. Set the **Database Type** to **Other**.
   d. Ensure that the **Database Driver** is set to **Other**, then click **Next**.
5. On the Transaction Options page, deselect **Supports Global Transactions**, then click **Next**.
6. On the Connection Properties page, follow these steps:
   a. For **Database Name**, enter the name of your database. If you are adding an Oracle BI EE data source, enter any non-empty string; the value does not matter because this property is not used for Oracle BI EE. If you are adding a Siebel Analytics Server data source, enter the *`SAS_DSN_name`*.

   Note: To add a Siebel Analytics Server data source, you must first create and configure a Siebel Analytics ODBC data source name (*`SAS_DSN_name`*). Refer to *Siebel Analytics Server Administration Guide* for complete details. The Siebel Analytics Server data source name cannot contain spaces.

   b. For **Host Name**, enter the name of the computer hosting the database server. If you are adding an Oracle BI EE or Siebel Analytics Server data source, enter the name of the computer hosting Oracle BI EE or Siebel Analytics Server.
   c. For **Port**, enter the port number on the database server used to connect to the database (such as `1433` for SQL Server, `1521` for Oracle Database, `50000` for DB2, or `9703` for Oracle BI EE and Siebel Analytics Server).
   d. For **Database User Name**, enter the name of the database run-time user. If you are adding an Oracle BI EE or Siebel Analytics Server data source, enter the name of an Oracle BI EE or Siebel Analytics Server user.

e. For **Password**, enter the password of the database run-time user. If you are adding an Oracle BI EE or Siebel Analytics Server data source, enter the password of the Oracle BI EE or Siebel Analytics Server user. Then, click **Next**.

7. On the Test Database Connection page, for **Driver Class Name**, enter the full package name of the JDBC driver class used to create the physical database connections in the connection pool (note that this driver class must be in the classpath of any server to which it is deployed):

   - **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`
   - **Oracle Database:** `oracle.jdbc.pool.OracleDataSource`
   - **DB2:** `com.ibm.db2.jcc.DB2Driver`
   - **Teradata:** `com.teradata.jdbc.TeraConnectionPoolDataSource`
   - **Oracle BI EE:** `oracle.bi.jdbc.AnaJdbcDriver`
   - **Siebel Analytics Server:** `sun.jdbc.odbc.JdbcOdbcDriver`

8. For URL, enter the URL of the database to which you want to connect. The format of the URL varies by data source type:

   - **SQL Server:** `jdbc:sqlserver://`*`db_host`*`:`*`db_port`*

     Note: If the database is on a SQL Server named instance, specify the *`db_host`* parameter using the format *`host_name\instance_name`*.

   - **Oracle Database:** `jdbc:oracle:thin:@`*`db_host`*`:`*`db_port`*`:`*`sid`*
   - **DB2:** `jdbc:db2://`*`db_host`*`:`*`db_port`*`/`*`db_name`*
   - **Teradata:** `jdbc:teradata://`*`server_name:port/db_name/param1,param2,...`*

     Note: If *`db_name`* is missing, the current login user's default database is used. For example, with default database **`RTD30`**, `jdbc:teradata://64.181.232.117//TMODE=ANSI,CHARSET=ASCII` executes as `jdbc:teradata://64.181.232.117/`**`RTD30`**`/TMODE=ANSI,CHARSET=ASCII`

   - **Oracle BI EE:** `jdbc:oraclebi://`*`server_name`*`:9703/user=`*`bi_user_name`*`;password=`*`bi_password`*`;catalog=`*`catalog_name`*`;`

     The catalog name is optional.

   - **Siebel Analytics Server:** `jdbc:odbc:`*`SAS_DSN_name`*

9. In the Properties field, enter properties and their values required by the JDBC driver. The properties you need to provide vary by data source type:

   - For **SQL Server**, enter the properties `user=`*`db_user_name`* and `DatabaseName=`*`db_name`* on separate lines
   - For **Oracle Database**, enter the property `user=`*`db_user_name`*
   - For **DB2**, enter the property `user=`*`db_user_name`*
   - For **Teradata**, enter the property `username=`*`db_user_name`*

- For **Oracle BI EE**, there are no required properties. Leave the **Properties** field blank.
- For **Siebel Analytics Server**, enter the property `user=`*`SAS_user_name`*

10. Scroll to the bottom of the page. For **Test Table**, enter the name of an existing table in the database.

> **Note:** Do not test the connection if you are adding an Oracle BI EE or Siebel Analytics Server data source. Instead, skip to Step 12.

11. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**.
12. Select the server where you want the changes to be made available (for example, **RTD_Server**). You *must* perform this step before completing the data source configuration.
13. Click **Finish**.
14. Click **Activate Changes**.

8.3.2.1 Creating an Oracle JDBC Provider for the Oracle RTD Database on WebLogic 11g 10.3.3+

To create a JDBC provider when your Oracle RTD database is in an Oracle database server, and Oracle RTD is deployed on WebLogic 11g 10.3.3+, perform the following steps:

1. Log into the WebLogic Server Administration Console with the administrator user name and password.
2. Navigate the path **Services** -> **Data Sources** -> **New** -> **Generic Data Source**.
3. On the JDBC Data Source Properties page, follow these steps:
   a. For **Name**, provide a descriptive name for the data source (for example, *`db_name`*`_DS`).
   b. For **JNDI Name**, enter the same value you provided for **Name**. This value will appear in Decision Studio when you perform an Import in a data source object.
   c. For **Database Type**, select **Oracle**.
   d. For the **Database Driver,** select **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later**, then click **Next**.
4. On the Transaction Options page, deselect **Supports Global Transactions**, then click **Next**.
5. On the Connection Properties page, follow these steps:
   a. For **Database Name**, enter the name of your 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**.

6. On the Test Database Connection page, leave all the settings already filled, except enter SDAPPS for **Test Table Name**.
7. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**.
8. Select the server where you want the data source to be made available (for example, **RTD_Server**). You must perform this step before completing the data source configuration.
9. Click **Finish**, then click **Activate Changes**.

8.3.3 Adding the New Data Source to Oracle RTD

After you create a data source in WebLogic, you must add the new data source to Oracle RTD. To do this, you must add a resource reference to the web.xml file within the soap.war and rtis.war archives, as follows:

1. Log into the WebLogic Server Administration Console for your Oracle RTD domain at the URL http://*weblogic_host*:*port*/console.
2. In the tree to the left under Domain Structure, navigate to Deployments.
3. Select OracleRTD under Deployments.
4. Select **Stop -> Force Stop Now**.
5. In the tree to the left under Domain Structure, navigate to Environment and then Servers.
6. Select your server, for example, RTD_Server, under Servers.
7. Select the Control tab.
8. Select the checkbox for your server, for example, RTD_Server, under Servers.
9. Select **Shutdown -> Force Shutdown Now**.
10. Go to the directory where you expanded the RTD.ear file during installation, for example, c:\bea\user_projects\domains\oraclertd_domain\servers\RTD_Server\stage\OracleRTD\expanded.
11. Open the rtis.war archive and 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_RTIS and paste it after the existing section. In the copied section, replace the string SDDS with the JNDI name you entered in Step 4 of Section 8.3.2. For example:

    ```
    <resource-ref id="jndi_name_RTIS">
      <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 rtis.war.
12. Open the soap.war archive and 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 for SDDS_Axis and paste it after the existing section. In the copied section, replace the string SDDS with the JNDI name you entered in Step 4 of Section 8.3.2. For example:

    ```
    <resource-ref id="jndi_name_Axis">
    ```

```
  <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 `soap.war`.

13. Return to the Administration Console.
14. In the tree to the left under Domain Structure, navigate to Environment and then Servers.
15. Select your server, for example, RTD_Server, under Servers.
16. Select the Control tab.
17. Select the checkbox for your server, for example, RTD_Server, under Servers.
18. Select **Start**.
19. In the tree to the left under Domain Structure, navigate to Deployments.
20. Select the checkbox next to OracleRTD under Deployments.
21. Select **Start -> Servicing all requests**.

8.4 Creating Additional JDBC Data Sources in JBoss

If you are running Real-Time Decision Server on JBoss 4.3 EAP, follow the steps in this section to configure JDBC data sources so that your Inline Services can access outside data.

This section contains the following topics:

- Section 8.4.1, "Creating a Connection Pool and Data Source in JBoss"
- Section 8.4.2, "Adding the New Data Source to Oracle RTD"
- Section 8.4.3, "Updating RTD.ear in JBoss"

8.4.1 Creating a Connection Pool and Data Source in JBoss

To create a connection pool and data source in JBoss, and to copy the data source to the JBoss deploy directory, perform the folloiwng steps:

1. In *`RTD_HOME`*`/package/jboss`, choose your supported database. Copy the file appropriate to that database and rename it, as follows:
   - For DB2:

     `rtd-db2-ds.xml -> rtd-custom-db2-ds.xml`
   - For Oracle:

     `rtd-oracle-ds.xml` -> `rtd-custom-oracle-ds.xml`
   - For SQL Server:

     `rtd-sqlserver-ds.xml` -> `rtd-custom-sqlserver-ds.xml`
2. Open the renamed file and replace the following:
   - `${DB_SERVER}` with your database server

     For example: `localhost`

> **Note:** If the database is on a SQL Server named instance, specify the name of your database server using the format *host_name\instance_name*.

- `${DB_PORT}` with your database port

  For example: `50000` for db2, `1521` for oracle, `1433` for sqlserver
- `${DB_NAME}` with your database name

  For example: `RTD` for db2 and sqlserver, `orcl` for oracle
- `${DB_USER}` with your database user name

  For example: `jsmith`
- `${DB_PASSWORD}` with your database user password
- `SDDS` (in `<jndi-name>SDDS</jndi-name>`) with a jndi name of your choice

  For example, the name of your database

3. Save the file (`rtd-custom-*-ds.xml`) and copy it to *JBOSS_HOME*`/jboss-as/server/default/deploy/`.

8.4.2 Adding the New Data Source to Oracle RTD

After you create a connection pool and data source in JBoss, you must add the new data source to Oracle RTD. To do this, you must add a resource reference to four different XML files. Before you begin, ensure that JBoss is not running.

To add the new data source to Oracle RTD:

1. Locate *RTD_HOME*`/package/RTD.ear` and make a backup copy. Then open `RTD.ear` with an archive utility such as WinZip.
2. In `RTD.ear`, locate and extract the file `./rtis.war`.
3. In `rtis.war`, locate and extract the file `./WEB-INF/web.xml`.
4. Open `web.xml` for editing and scroll to the bottom of the file. Copy the section for the definition of the resource reference of `SDDS_RTIS` and paste it after the existing section. In the copied section, replace the string `SDDS` with the JNDI name (*jndi_name*) that you entered in Step 2 of Section 8.4.1. For example:

   ```
   <resource-ref id="jndi_name_RTIS">
   <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>
   ```

   Then, save the changes and close the file.
5. In `rtis.war`, locate and extract the file `./WEB-INF/jboss-web.xml`.
6. Open `jboss-web.xml` for editing and scroll to the bottom of the file. Copy the section for the definition of the resource reference of `SDDS` and paste it after the existing section. In the copied section, replace the string `SDDS` with the JNDI name (*jndi_name*) that you entered in Step 2 of Section 8.4.1. For example:

   ```
   <resource-ref>
   ```

```
<res-ref-name>jndi_name_RTIS</res-ref-name>
<res-type>javax.sql.DataSource</res-type>
<jndi-name>jndi_name</jndi-name>
</resource-ref>
```

Then, save the changes and close the file.

7. Using your archive utility, copy `web.xml` and `jboss-web.xml` back into `rtis.war` at `./WEB-INF/` (replacing the original files). Save `rtis.war`.
8. Using your archive utility, copy `rtis.war` back into `RTD.ear` at `./` (replacing the original file). Save `RTD.ear`.
9. In `RTD.ear`, locate and extract the file `./soap.war`.
10. In `soap.war`, locate and extract the file `./WEB-INF/web.xml`.
11. Open `web.xml` for editing and scroll to the bottom of the file. Copy the section for the definition of the resource reference of `SDDS_Axis` and paste it after the existing section. In the copied section, replace the string `SDDS` with the JNDI name (*`jndi_name`*) that you entered in Step 2 of Section 8.4.1. For example:

   ```
   <resource-ref id="jndi_name_Axis">
   <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>
   ```

   Then, save the changes and close the file.
12. In `soap.war`, locate and extract the file `./WEB-INF/jboss-web.xml`.
13. Open `jboss-web.xml` for editing and scroll to the bottom of the file. Copy the section for the definition of the resource reference of `SDDS` and paste it after the existing section. In the copied section, replace the string `SDDS` with the JNDI name (*`jndi_name`*) that you entered in Step 2 of Section 8.4.1. For example:

   ```
   <resource-ref>
   <res-ref-name>jndi_name_Axis</res-ref-name>
   <res-type>javax.sql.DataSource</res-type>
   <jndi-name>jndi_name</jndi-name>
   </resource-ref>
   ```

   Then, save the changes and close the file.
14. Using your archive utility, copy `web.xml` and `jboss-web.xml` back into `soap.war` at `./WEB-INF/` (replacing the original files). Save `soap.war`.
15. Using your archive utility, copy `soap.war` back into `RTD.ear` at `./` (replacing the original file). Save `RTD.ear`.

8.4.3 Updating RTD.ear in JBoss

To complete the process of adding the data source to Oracle RTD and JBoss, copy the updated `RTD.ear` file to the JBoss deploy directory when JBoss has been stopped, as follows:

1. Stop JBoss.
2. Copy the updated `RTD.ear` in *`RTD_HOME`*`/package/` to *`JBOSS_HOME`*`/jboss-as/server/default/deploy/`, replacing the existing file.
3. Start JBoss.

For information about starting and stopping JBoss on your operating system, see the Note - which refers to *both* start and stop scripts - at the end of the section appropriate to your installation:

- Section 6.1.4.1, "Modifying the Windows JBoss Start Script"
- Section 6.1.4.2, "Modifying the Unix JBoss Start Script"

8.5 Testing a New Enterprise Data Source

After you add a new enterprise data source, follow the steps in this section to ensure the data source is configured properly. Before you begin, ensure that Oracle RTD is started.

To test a new enterprise data source:

1. Start Decision Studio by running `eclipse.exe` in *RTD_HOME*`\eclipse`. Then, create a new Inline Service, or open an existing Inline Service. See the Decision Studio Help for more information about how to do this.
2. Expand the **Service Metadata** folder, then right-click **Data Sources** and select **New SQL Data Source**.
3. For **Display Label**, enter a name for the data source you want to test, then click **OK**.
4. Click **Import**. Then, select the data source you want to test from the **JDBC Data Source** drop-down list. The list of Tables and Views is updated with tables and view names from that data source.
5. Select a particular table or view, then click **Finish** in the Import dialog box. The list of available columns appears in the Output table in the Data Source editor.
6. Write some basic code to ensure that the actual rows are retrieved from the tables at run time.

If you were not able to complete any of the preceding steps, check your data source configuration settings and try again.

9

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 9.1, "Decision Center Browser Configuration"
- Section 9.2, "Real-Time Decision Server Logging Configuration"
- Section 9.3, "Accessing Oracle Real-Time Decisions Client Tools"

9.1 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.

9.2 Real-Time Decision Server Logging Configuration

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 15.1, "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.

9.3 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 15.1, "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 9.3.1, "Accessing Decision Studio"
- Section 9.3.2, "Accessing Decision Center"
- Section 9.3.3, "Accessing Load Generator"

9.3.1 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`.

9.3.2 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`.

9.3.3 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`.

10

Production Deployment of Oracle Real-Time Decisions

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 an all-in-one configuration, or a full deployment configuration. System configuration may need to be adjusted based upon specific customer requirements and application characteristics.

This chapter contains the following topics:

- Section 10.1, "All-In-One Configuration"
- Section 10.2, "Full Deployment Configuration"

10.1 All-In-One Configuration

An All-In-One configuration supports a proof of concept or pilot-type implementation. The intent 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. The configuration supports approximately 100 account managers, or about 50 call center agents.

Table 10–1 lists the hardware requirements for an All-In-One configuration.

Table 10–1 Hardware Requirements for All-In-One Configuration

Hardware	Minimum Configuration
J2EE Application Server	WebLogic
Operating System	Windows
Number of CPUs	2
CPU	2+ GHz
RAM	2 GB
Hard Drive	100 GB
Database	SQL Server, Oracle Database, or DB2

10.2 Full Deployment Configuration

The Full Deployment configuration supports an implementation of approximately 1500 B2B Account Managers or 1000 call center agents in a production environment. 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, and the number of data sources. The Oracle Corporation Professional Services Group can assist in determining the most appropriate configuration for your business and technical requirements.

Table 10–2 lists the hardware requirements for a Full Deployment Configuration.

Table 10–2 Hardware Requirements for Full Deployment Configuration

Server	Component	Minimum Configuration
Decision Servers (2)	CPU	2 x 2-CPU, 2+GHz, 2GB RAM
	J2EE Application Server	Standalone OC4J, Oracle Application Server, WebSphere, or WebLogic
	Operating System	Windows, Solaris, AIX, HP-UX, or Linux
Database Server[1]	CPU	4-CPU, 2+GHz, 8GB RAM
	DBMS	SQL Server, Oracle Database, or DB2
Learning Server[2]	CPU	2-CPU, 2+GHz, 2GB RAM
	J2EE Application Server	Standalone OC4J, Oracle Application Server, WebSphere, or WebLogic
	Operating System	Windows, Solaris, AIX, HP-UX, or Linux

[1] The ratio of database server CPUs to Real-Time Decision Server CPUs is approximately 1:1.

[2] The ratio of Learning Server CPUs to Real-Time Decision Server CPUs is approximately 1:4.

11

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 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-3.0.0.`*`xxx`*`.zip`, in the directory *`RTD_HOME`*`\client\CommandLineDeploy`.

First, unzip the file `rtd-deploytool-3.0.0.`*`xxx`*`.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-3.0.0.`*`xxx`*`.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.

11.1 Deploying the Inline Service

To deploy the Inline Service from the command line, perform the following steps:

1. Navigate to the directory containg 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 11–1.

Table 11–1 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 11–2.

Table 11–2 Command Line Deployment Positional Options

Parameter	Description *(includes references to Table 11–1 Format entries)*
username	Name of a user logging in to the RTD server.
password	The user’s password. For a blank password, specify "".

Table 11–2 (Cont.) Command Line Deployment Positional Options

Parameter	Description *(includes references to Table 11–1 Format entries)*
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
```

12

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 Snaphots, 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 12.1, "Overview of Setting Up and Using Model Snapshots"
- Section 12.2, "Model Snapshot Tables Schema"
- Section 12.3, "Configuring the Model Snapshot Tables"
- Section 12.4, "Populating and Clearing the Model Snapshot Tables"
- Section 12.5, "Creating Reports from the Model Snapshot Data"
- Section 12.6, "Handling Partitions"
- Section 12.7, "Tuning the Model Snapshot Process"

12.1 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.

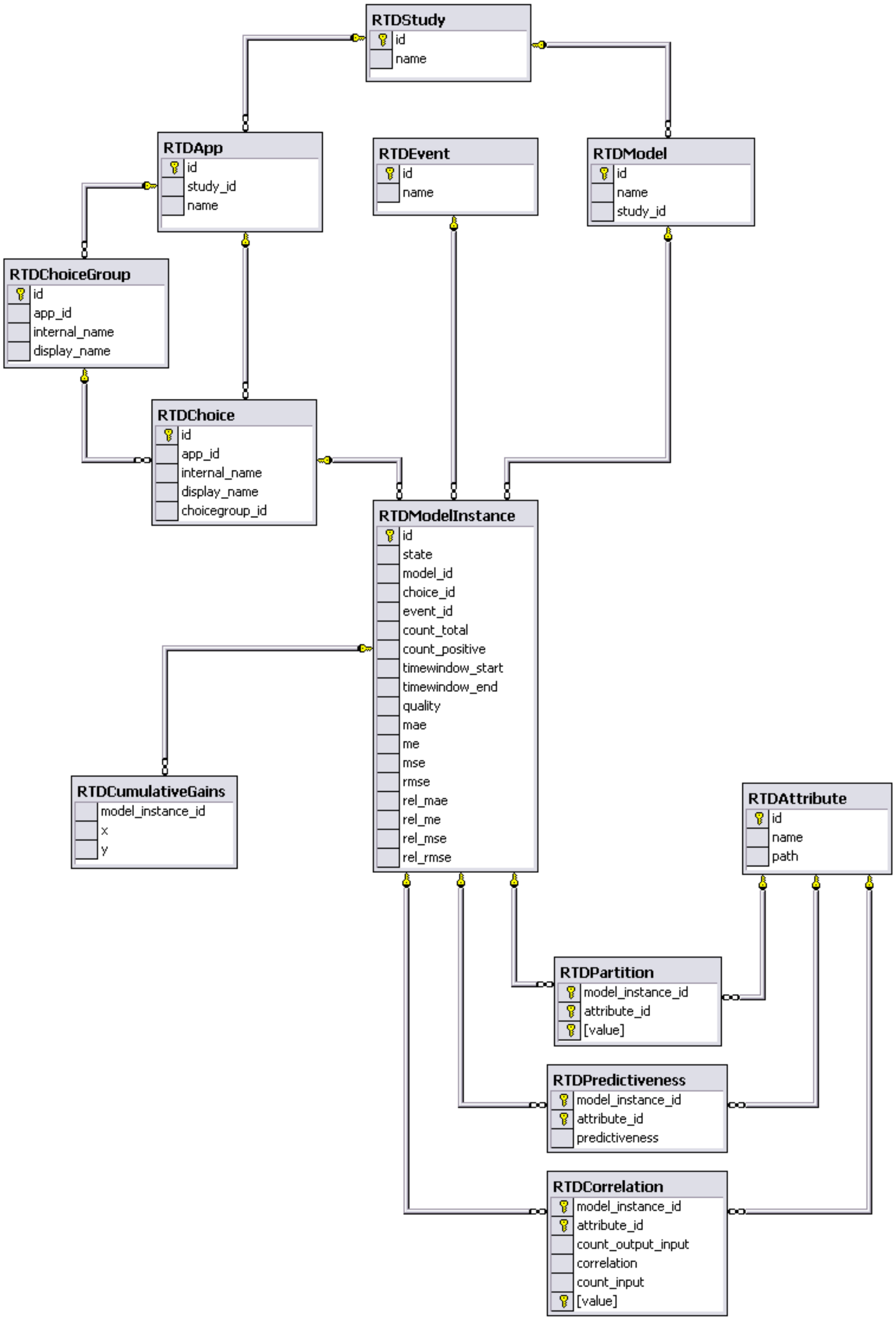
There are two parameters related to model snapshots that are for tuning purposes. For more information, see Section 12.7, "Tuning the Model Snapshot Process."

12.2 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 12–1, which shows the model snapshot tables, 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 12–1 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):

- 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.
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.

Column	Description
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.
time_window_start	Start of time window.
time_window_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.
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.

12.3 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.

Parameter	Description
db_port	The database port number.
db_name	The name of the database or, for Oracle Database, the SID.
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 Chapter 8, "Configuring Data Access for Oracle Real-Time Decisions."

   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 OC4J, continue at step 5.
   - If your application server is WebSphere, continue at step 11.
   - If your application server is WebLogic, continue at step 14.

   Adding the Data Source to Oracle RTD in OC4J

5. If Oracle RTD is running on standalone OC4J, go to the directory *OC4J_HOME*/j2ee/home/applications/OracleRTD. If Oracle RTD is running on Oracle Application Server, go to *ORACLE_AS_HOME*/j2ee/*oc4j_instance*/applications/OracleRTD.

   This directory is the location where the RTD.ear file was expanded when you deployed it as an application.

6. Locate the file ./ls/WEB-INF/web.xml and open it for editing. Scroll to the bottom of the file. Copy the section for the definition of the resource reference of SDDS and paste it after the existing section. In the copied section, replace the string SDDS with the JNDI name (*jndi_name*) that you entered 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>
   ```

   Then, save the changes and close the file.

7. Go to the directory *OC4J_HOME*/j2ee/home/application-deployments/OracleRTD. Make sure to go to the application-deployments directory, not the applications directory.

8. Locate the file `./ls/orion-web.xml` and open it for editing. Scroll to the bottom of the file. Copy the line for the definition of the resource reference mapping of `SDDS` and paste it after the existing line. In the copied line, replace the string `SDDS` with the JNDI name (*`jndi_name`*) that you entered in step 3. For example:

   ```
   <resource-ref-mapping name="jndi_name" location="jndi_name"/>
   ```

   Then, save the changes and close the file.

9. Start OC4J.

10. Continue at step 16.

    Adding the Data Source to Oracle RTD in WebSphere

11. 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
    ```

12. 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.

13. Continue at step 16.

    Adding the Data Source to Oracle RTD in WebLogic

14. Go to the directory where you expanded the `RTD.ear` file during installation (*`RTD_HOME`*`/package/expanded`).

15. 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 16.

16. 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.

12.4 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**

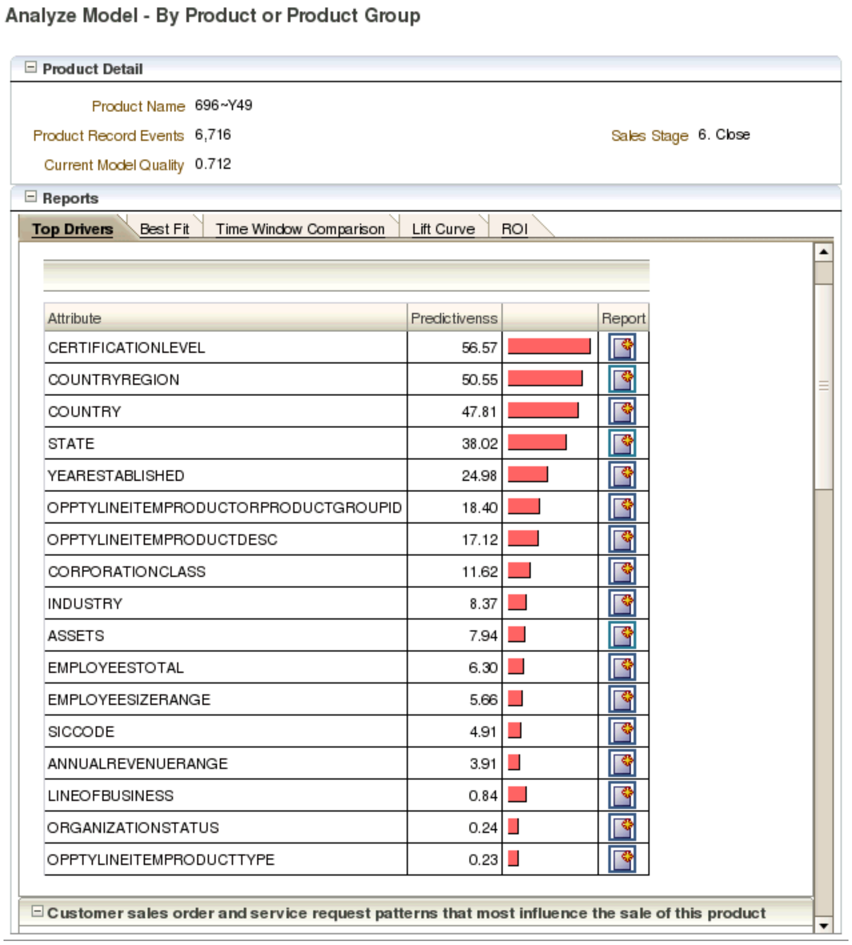
      Deletes all previously snapped data for the Study, and rewrites all Study data, up to the current time.
    - **IncrementalSnapshot**

      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.

12.5 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.

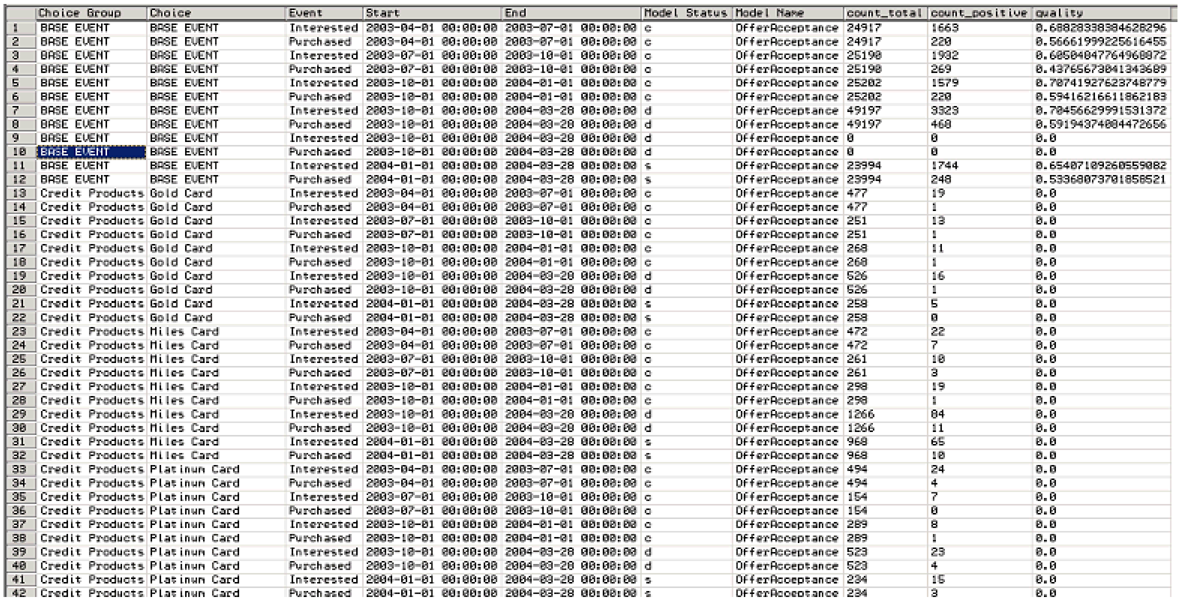
12.5.1 Counts by Choice Query

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 12–2 shows the results of the Counts by Choice query.

Figure 12–2 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	25190	1932	0.68504047764960872
4	BASE EVENT	BASE EVENT	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	25190	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.

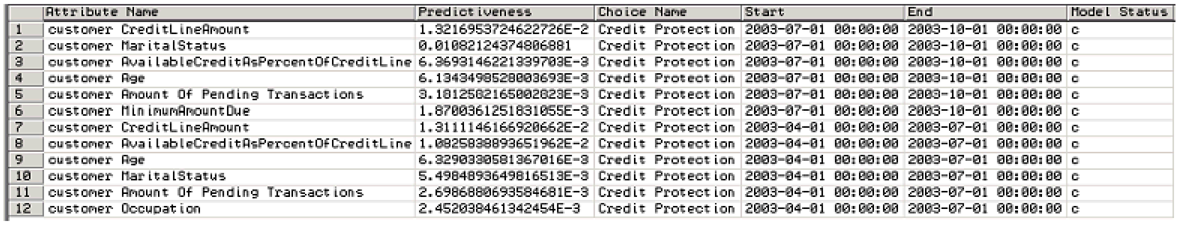
12.5.2 Top Six Predictive Attributes Query

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 12–3 shows the results of the Top Six Predictive Attributes query.

Figure 12–3 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	1.0825838893651962E-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

12.5.3 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.

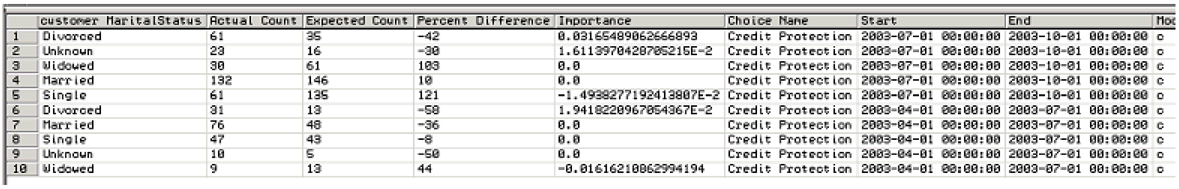
```
select cor.value               'customer MaritalStatus',
       cor.count_output_input  'Actual Count',
       mi.count_positive*cor.count_input/mi.count_total 'Expected Count',
  100* (  (mi.count_total*cor.count_output_input)
        / (mi.count_positive*cor.count_input) - 1) 'Percent Difference',
       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 12–4 shows the results of the Difference Between Expected and Actual Counts query.

Figure 12–4 Difference Between Expected and Actual Counts Query Results



	customer MaritalStatus	Actual Count	Expected Count	Percent Difference	Importance	Choice Name	Start	End	Moc
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	-0.01616210062994194	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.

12.6 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

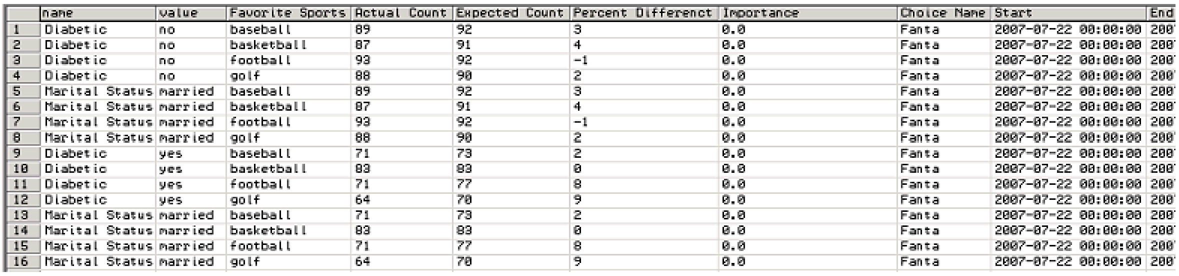
Whether or not a Model is partitioned can influence the results of queries on the model snapshot tables. To avoid repetitions in your results, include RTDPartition and RTDAttribute join conditions in your query.

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 12–5 shows the results of the Partitioned Expected and Actual Counts query.

Figure 12–5 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

12.7 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 15.3.2, "About OracleRTD > SDClusterPropertyManager > Misc."

13 Deploying Oracle RTD into an Oracle RTD Cluster

Oracle RTD may be deployed into a cluster to achieve any of the following objectives:

- Increased processing power
- Increased memory, to accommodate more concurrent sessions
- Increased availability in the event of hardware failures

You can deploy Oracle RTD to several application server types. This chapter describes in general how to deploy Oracle RTD into a clustered J2EE environment. It presents an outline of the concepts and processes for implementors who are familiar with deploying J2EE applications on application servers.

> **Note:** This chapter does not describe how to set up the HTTP server that will be the load balancing front-end to the cluster. The HTTP server provides the publicly accessible URLs to the Oracle RTD services, but there are no special considerations about the Oracle RTD installation in that regard.

For convenience, the following abbreviations are used in this chapter to describe the different Oracle RTD services:

- **DS** - Decision Service
- **LS** - Learning Service
- **DC** - Decision Center

This chapter contains the following topics:

- Section 13.10, "Setting up a JBoss Cluster"

13.1 Overview of Oracle RTD Clustering

Oracle RTD is deployed through a J2EE application archive file, `RTD.ear`, which contains multiple web applications corresponding to Decision Service (DS), Learning Service (LS), and Decision Center (DC). In a production environment, DS is typically deployed into separate cluster nodes (Oracle RTD instances) from DC and LS to prevent the activities of DC or LS from impacting the DS response times.

Two combinations of Oracle RTD services are common:

- **Oracle RTD Cluster Type 1**

  DS, DC, and LS all deployed into every Oracle RTD instance

- **Oracle RTD Cluster Type 3**

  DS deployed by itself, DC and LS deployed together

Two further cluster combinations are possible, but they are less commonly used:

- **Oracle RTD Cluster Type 2**

  DS and DC deployed together in some Oracle RTD instances

  LS deployed by itself in other instances.

- **Oracle RTD Cluster Type 4**

  DS, DC, and LS each deployed into separate Oracle RTD instances

From a clustering perspective, it is important to note that Oracle RTD is delivered as a single J2EE `ear` file. In some application server types, the granularity of deployment is the `ear` file, and not the contained Web Application Archives (`war` files). In these cases, all web applications within the enterprise application get deployed together. This chapter describes how to work around this packaging issue in order to deploy individual services of Oracle RTD into separate cluster nodes.

This chapter also describes several configuration parameters that are required when deploying Oracle RTD into a cluster. These parameters can be ignored in non-clustered environments.

Certain cluster-specific behaviors are also described, such as the role of the Oracle RTD cluster coordinator in managing cluster singletons. The cluster coordinator, for example, ensures that Learning Service runs on precisely one Oracle RTD instance at a time, restarting it on another instance should the first one fail.

13.2 Terminology

The section describes both general terms and terms that have a particular meaning and usage for Oracle RTD clustering.

- **Application Server, Server Instance, or AS Instance**

  In this chapter, these three terms are equivalent to a Java Virtual Machine on which a J2EE application, such as Oracle RTD, may be deployed. Multiple AS instances may reside on the same host machine.

Note: This definition of an AS instance is different from the definition described by Oracle Application Server (OAS). In OAS, an OAS instance can have multiple JVMs, a situation that is not supported by Oracle RTD.

- **Oracle RTD Instance**

  This is an AS instance onto which the Oracle RTD application has been deployed.

- **Oracle RTD Cluster and J2EE Cluster**

  There is an important distinction between these two terms. For details, see Section 13.2.1, "Distinguishing Oracle RTD Clusters from J2EE Clusters."

- **Oracle RTD Cluster Coordinator**

  One Oracle RTD instance, the first one to announce its presence, is designated at runtime to be the Oracle RTD cluster coordinator. The coordinator monitors cluster singletons (for example, Learning Service), to ensure that precisely one deployed singleton is running at a time.

  You can tell which is the current coordinator by searching the `server.log` file of each Oracle RTD instance. Search backwards from the end of the file for the string "CLUSTER COORDINATOR."

  It is possible for an Oracle RTD cluster to temporarily have more than one cluster coordinator, as in the case of a temporary network partitioning due to network communication failures or timeouts. Within a minute or so of the network coming back together, the multiple coordinators should relinquish their empires to a single coordinator.

- **Oracle RTD Home**

  This is the directory into which Oracle RTD was installed. It contains not only `RTD.ear` (the J2EE application archive), but also sample code and required libraries.

  A typical value for Oracle RTD Home on a Microsoft Windows computer is `C:\OracleBI\RTD`. In this chapter, Oracle RTD Home is also referred to as *RTD_HOME*.

- **Oracle RTD Runtime Home**

  This is the directory in which the application server runs Oracle RTD. Although its location is application server type-specific, generally it will have the same structure as an expanded version of `RTD.ear`.

  In this chapter, references to Oracle RTD Runtime Home appear as *RTD_RUNTIME_HOME*.

  The following list shows some application server specific generic names and typical examples of *RTD_RUNTIME_HOME*:

  – **Oracle AS 3.3 generic names**

    ORACLE_HOME`\j2ee\`OC4J_INSTANCE_NAME`\applications\`*APP_NAME*

    Where:

    ORACLE_HOME = `C:\oracle\product\as-33`

    OC4J_INSTANCE_NAME = `rtd1_2`

APP_NAME = `OracleRTD`

- **WebLogic generic names**

  `MW_HOME\user_projects\domains\DOMAIN\servers`
  `\SERVER_NAME\stage\APP_NAME\expanded`

  Where:

  MW_HOME = `C:\Oracle\Middleware`

  DOMAIN = `oraclertd_domain`

  SERVER_NAME = `RTDServer1`

  APP_NAME = `OracleRTD`

- **WebSphere 6.1 generic names**

  WS_HOME\AppServer\profiles\SERVER_NAME
  `\installedApps\host_nameNode_name\OracleRTD.ear`

  Where:

  WS_HOME = `C:\Program Files\IBM\WebSphere`

  SERVER_NAME = `AppSrv01`

13.2.1 Distinguishing Oracle RTD Clusters from J2EE Clusters

The term "cluster" needs some clarification, because Oracle RTD and each application server system, Oracle AS, WebSphere, and WebLogic, all define and use the same term in different ways.

This section defines the terms ***Oracle RTD cluster*** and ***J2EE cluster***, and compares these definitions to the other application-server specific definitions.

- **Oracle RTD Cluster**

  From the perspective of Oracle RTD, a cluster is the entire set of Oracle RTD instances running any combination of Oracle RTD services, DS, DC, or LS.

  The Oracle RTD instances of an Oracle RTD cluster all share the same SDDS database, and all have the same SDGroupName setting. These two properties are really what determines which Oracle RTD instances are in an Oracle RTD cluster.

  A Type 1 cluster is an Oracle RTD cluster where DS, DC, and LS all are enabled in every Oracle RTD instance.

  A Type 3 cluster, the main focus of this chapter, is an Oracle RTD cluster where DS runs in separate Oracle RTD instances from the instances running DC and LS.

- **OAS Cluster**

  To Oracle Application Server, a cluster is the set of OAS instances managed by the same Enterprise Manager application, `ascontrol`. J2EE applications are not necessarily deployed uniformly to all the OAS instances of an OAS cluster.

- **OC4J Group**

  In OAS, the OAS instances used by Oracle RTD run OC4J (Oracle J2EE Container For Java), and are known as OC4J instances.

  OAS uses the term "OC4J group" for a collection of OC4J instances into which applications are deployed uniformly. For example, deploying Oracle RTD into an OC4J group automatically deploys it into all the OC4J instances of the OC4J group.

Deploying an Oracle RTD Type 3 cluster into OAS involves a single OAS cluster and two OC4J groups, because Oracle RTD will be deployed twice, once into each OC4J group.

- **WebSphere Cluster**

  In WebSphere, a cluster is similar to an OC4J group, in that an application is deployed uniformly to all instances of the cluster.

  Deploying an Oracle RTD Type 3 cluster into WebSphere involves two WebSphere clusters, because Oracle RTD will be deployed twice, once into each WebSphere cluster.

- **WebLogic Cluster**

  A WebLogic cluster is similar to a WebSphere cluster, in that an application is deployed uniformly to all instances of the cluster.

  Deploying an Oracle RTD Type 3 cluster into WebLogic involves two WebLogic clusters, because Oracle RTD will be deployed twice, once into each WebLogic cluster.

- **J2EE Cluster**

  For consistency and ease of discussion, a new term, "J2EE cluster", is used in this chapter, which means any one of the following: "WebSphere cluster", "WebLogic cluster", or "OC4J Group". An application deployed to a J2EE cluster is deployed uniformly to all instances of the cluster.

  Note that the main two terms used in this chapter are "Oracle RTD cluster" and "J2EE cluster."

  An Oracle RTD Type 3 cluster contains two J2EE clusters.

  Oracle RTD is deployed differently into each of the two J2EE clusters in order to activate different Oracle RTD services within the two J2EE clusters.

  This compares to an Oracle RTD Type 1 cluster, which contains just one J2EE cluster, because Oracle RTD is deployed uniformly across all Oracle RTD instances.

13.3 Cluster-Specific Configuration Properties

This section describes the system properties and Oracle RTD configuration settings that are typically necessary in a clustered deployment. Instructions for setting them appear in subsequent sections.

- **SDGroupName**

  This is the name of the Oracle RTD cluster. This is typically set via a system property in the JVM (Java Virtual Machine), or it can be set via the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**. The default value is the name of the containing host, so it must be changed when multiple hosts are included in the cluster.

  Each cluster must have a unique SDGroupName, in order to avoid communication conflicts with other Oracle RTD installations in the same local area network.

  Each node (Oracle RTD Instance) in the cluster must have the same SDGroupName, and they must all access the same SDDS data source, where configuration data and analytic models reside.

- **rtd.instanceName**

This is a system property set in the JVM to specify the unique name for the Oracle RTD instance within the cluster. The default value **RTDServer** is unsuitable in a cluster containing more than one Oracle RTD instance, because it would not be unique.

The format of the name, which must not include a space, follows file name conventions. This is because sometimes the instance name is used to distinguish log files and configuration files when multiple Oracle RTD instances are deployed to the same host.

Because Oracle RTD uses system properties, only one Oracle RTD instance can run in a given JVM.

- **rtd.instancesShareRuntimeHome**

  This is a system property set in the JVM. Its default value is **false**, but must be set to **true** in installations (typically only WebSphere) where multiple Oracle RTD instances share the same runtime home directory. When **true**, the Oracle RTD instance name is used as a directory name to isolate the log files, configuration cache files, and Inline Service metadata cache files for the instances on this host.

  Table 13–1 shows how the values of **rtd.instancesShareRuntimeHome** control the locations of the Oracle RTD system files. In this table,

  – *RTD_RUNTIME_HOME* stands for Oracle RTD Runtime Home, the directory in which the application server runs Oracle RTD

  – *RTD_INST_NAME* stands for the value of the rtd.instanceName system property

Table 13–1 Locations of Oracle RTD System Files

rtd.instancesShareRuntimeHome	Value=False	Value=True
Location of log file `server.log`	`RTD_RUNTIME_HOME/log/`	`RTD_RUNTIME_HOME/log/RTD_INST_NAME`
Location of config cache files: `sdconfig-machine.xml`, `sdconfig-cluster.xml`	`RTD_RUNTIME_HOME/etc/`	`RTD_RUNTIME_HOME/etc/RTD_INST_NAME`
Location of Inline Service metadata cache files	`RTD_RUNTIME_HOME/apps/`	`RTD_RUNTIME_HOME/apps/RTD_INST_NAME`

- **DecisionServiceAddress**

  This is the cluster-wide prefix of the URL to the Decision Service, an address that typically corresponds to a virtual address managed by a load balancer.

  The default value, **http://myHost:8080**, serves merely as an example. This must be changed when Decision Center is not co-located with Decision Service, in order to enable Decision Center's Interactive Integration Map to send test events to Decision Service. This attribute is accessed in the MBean, **OracleRTD > SDClusterPropertyManager > Misc**.

- **DecisionServiceEnabled**

  This can be set in a system property in the JVM options of the application server, or in the MBean, **OracleRTD > SDPropertyManager > Misc**.

  The default value is **true**, but must be set to **false** in any Oracle RTD instances where you do not want Decision Service to run. Setting this to **false** is just one of the required steps to disable DS. See Section 13.5, "Deploying Oracle RTD into an Oracle RTD Type 3 Cluster."

- **LearningServiceEnabled**

This can be set in a system property in the JVM options of the application server, or in the MBean, **OracleRTD > SDPropertyManager > Misc.**

The default value is **true**, but must be set to **false** in any Oracle RTD instances where you do not want Learning Service to run. Learning Service can be enabled in multiple Oracle RTD instances, but will run in only one at a time. Setting this to **false** is just one of the required steps to disable LS. See Section 13.5, "Deploying Oracle RTD into an Oracle RTD Type 3 Cluster."

- **JGroupsMulticastPort**

  This can be set in a system property in the JVM options of the application server, or in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**. The value must be the same in all Oracle RTD instances in the cluster.

  The default value is **45566**, and does not need to be changed unless other (non-Oracle RTD) applications on the local intranet are using the same multicast address (see JGroupsMulticastAddress) and multicast port.

  If other clusters of Oracle RTD are installed and use the same multicast address and port, each Oracle RTD cluster must use a different SDGroupName and the versions of Oracle RTD must be wire-compatible.

  If an old version of Oracle RTD is present whose group-communication messages are incompatible, the `server.log` file for Oracle RTD will register error messages concerning each receipt of an incompatible message. These log entries are benign, but will tend to clog up the log file. These incompatibilities are not usually noticed because the default settings of Oracle RTD change between releases when necessary to avoid conflict.

- **JGroupsMulticastAddress**

  This can be set in a system property in the JVM options of the application server, or in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**.

  The value must be the same in all Oracle RTD instances of the cluster. The default value is **228.64.16.34**, and does not need to be changed unless other (non-Oracle RTD) applications on the local intranet are using the same multicast address and multicast port. See more detailed discussion under JGroupsMulticastPort.

- **JGroupsDSMulticastPort**

  This is similar to JGroupsMulticastPort in terms of its location and scope. It is used by Decision Service in managing session affinity, when it receives a request whose session is being hosted on another Oracle RTD instance. DS uses the channel specified by JGroupsDSMulticastAddress and JGroupsDSMulticastPort to monitor and, if necessary, forward requests to other Oracle RTD instances running Decision Service.

  The value must be the same in all Oracle RTD instances of the cluster, and must be different from the value of JGroupsMulticastPort. The default value is **45567**.

- **JGroupsDSMulticastAddress**

  This is similar to JGroupsMulticastAddress in terms of its location and scope. It is used by Decision Service in managing session affinity, when it receives a request whose session is being hosted on another Oracle RTD instance. DS uses the channel specified by JGroupsDSMulticastAddress and JGroupsDSMulticastPort to monitor and, if necessary, forward requests to other Oracle RTD instances running Decision Service.

  The value must be the same in all Oracle RTD instances of the cluster, and does not need to be different from the value of JGroupsMulticastAddress as long as the combination of JGroupsMulticastAddress/JGroupsMulticastPort is different from the combination of JGroupsDSMulticastAddress/JGroupsDSMulticastPort. The default value is **228.64.16.34**, the same as the default value for JGroupsMulticastAddress.

- **RestrictClusterMembers**

  When RestrictClusterMembers is **true**, Oracle RTD only communicates with Oracle RTD instances whose host addresses have been specified in the property TrustedClusterMembers. The default value is **true**.

  This property must have the same value across the cluster. It can be set as a system property, or in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**.

- **TrustedClusterMembers**

  This is a string containing a list of hosts that may participate in the cluster. Messages from Oracle RTD instances on hosts not listed here, if RestrictClusterMembers is true, will generate authentication errors in `server.log`.

  The host addresses are IP addresses, not host names. Do not include port numbers in these addresses, because port assignments are dynamically allocated and hence cannot be pre-configured. Entries must be separated by a semicolon (";"). The default value is an empty string.

  This property must have the same value across the cluster. It can be set as a system property, or in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**.

- **RestrictDSClients**

  When **true**, Decision Service only accepts requests from host addresses that have been specified in the property TrustedDSClients. The default value is **true**.

  This property must have the same value across the cluster. It can be set as a system property, or in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**.

- **TrustedDSClients**

  This is a string containing a list of hosts that may send Decision Service requests to Oracle RTD. If RestrictDSClients is **true** (the default value), then DS requests from hosts not listed here and not part of the Oracle RTD cluster will be rejected by DS and will generate authentication errors in `server.log`.

  The host addresses are IP addresses, not host names. Ports may optionally be included, separated from the address by a colon (":"). Entries are separated by semicolons (";").

  This property must have the same value across the cluster. It can be set as a system property, or in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**.

13.4 Creating Server Instances for Oracle RTD Type 3 Clusters

The main objective of an Oracle RTD Type 3 cluster is to have Decision Service running in separate Oracle RTD instances from the instances running Decision Center and Learning Service. This section describes how to set up the application server instances (which will become Oracle RTD instances after deployment) in an Oracle RTD Type 3 cluster, prior to deploying Oracle RTD.

This section contains the following topics:

- Section 13.4.1, "Creating J2EE Clusters"

- Section 13.4.2, "Setting JVM Options"
- Section 13.4.3, "Setting Up JConsole Access"
- Section 13.4.4, "Setting Up JDBC Resources"

13.4.1 Creating J2EE Clusters

For an Oracle RTD Type 3 cluster, create two J2EE clusters, for example, **rtdGroup1** and **rtdGroup2**.

For an Oracle RTD Type 1 cluster, create one J2EE cluster only.

Note: IrtdGroup1 and rtdGroup2 are *J2EE clusters*, not Oracle RTD clusters.

Create and assign AS instances to the J2EE clusters. In this example, create a total of 4 AS instances, in 4 host machines, as in Table 13–2 and Table 13–3:

Table 13–2 J2EE Cluster rtdGroup1

AS Instance	Machine
rtd_1_1	host1
rtd_1_2	host2

Table 13–3 J2EE Cluster rtdGroup2

AS Instance	Machine
rtd_2_1	host3
rtd_2_2	host3

Note: There are no assumptions in this chapter about the number of machines involved.

Typically separate hosts are used to improve availability in the face of hardware failures, as well as to increase the total processing power.

Multiple instances on the same host can be used to work around the memory limitations of a 32-bit operating system on machines with greater than 2GB of memory.

13.4.2 Setting JVM Options

This section describes the JVM options that may be used for each Oracle RTD instance.

This section contains the following topics:

- Section 13.4.2.1, "Oracle RTD Clustering System Properties"
- Section 13.4.2.2, "Generic JVM Options"
- Section 13.4.2.3, "Application Server Specific JVM Options"

13.4.2.1 Oracle RTD Clustering System Properties

In general, system properties can be used for any Oracle RTD properties supported by the OracleRTD JMX MBeans, as long as they do not need to be subsequently changed by JMX. These system properties are usually used only for settings that either do not need to be changed via JMX, or that need to be set before JMX will work. For more information, see Section 13.3, "Cluster-Specific Configuration Properties."

Table 13–4 shows the Oracle RTD settings that are usually configured through system properties. In this table, the Scope values are defined as follows:

- RTD: Global to the entire Oracle RTD cluster
- J2EE: Global to the J2EE cluster
- AS: Specific to the AS instance

Table 13–4 Oracle RTD Settings Usually Configured Through System Properties

Name	Example Value	Scope	Description
SDGroupName	MyRTDCluster	RTD	N/A
rtd.instanceName	rtd_1_1	AS	Must be unique within the Oracle RTD cluster.
rtd.instancesShareRuntimeHome	false	RTD	false for OAS and WebLogic; true for WebSphere.
DecisionServiceEnabled	true	J2EE	Assuming an Oracle RTD Type 3 cluster: true, for rtdGroup1; false for rtdGroup2. As an alternative to setting this in system properties, it can be set via JConsole, as described in Section 13.5.3, "Disabling Certain Oracle RTD Services." It probably makes sense to set them as system properties, if adding an Oracle RTD instance requires copying the system properties setup anyway.
LearningServiceEnabled	false	J2EE	Assuming an Oracle RTD Type 3 cluster: false, for rtdGroup1; true for rtdGroup2. As an alternative to setting this in system properties, it can be set via JConsole, as described in Section 13.5.3, "Disabling Certain Oracle RTD Services." It probably makes sense to set them as system properties, if adding an Oracle RTD instance requires copying the system properties setup anyway
JGroupsMulticastAddress, JGroupsMulticastPort, JGroupsDSMulticastAddress, JGroupsDSMulticastPort	N/A	RTD	These do not usually need to be changed from their default values. If needed, it is better to set them as JVM system properties rather than through JConsole, because they are needed by JConsole.
RestrictClusterMembers	false	RTD	Set this to false to facilitate adding servers to the Oracle RTD cluster. If true (its default value), see Section 13.5.5, "Adding Restricted Cluster Members."

Table 13–4 (Cont.) Oracle RTD Settings Usually Configured Through System Properties

Name	Example Value	Scope	Description
TrustedClusterMembers	N/A	RTD	When RestrictClusterMembers is true (its default value), the set of allowed cluster hosts can be set here as a system property. However, it is easier to add new hosts if TrustedClusterMembers is managed through JMX instead of as a system property. See Section 13.5.5, "Adding Restricted Cluster Members."
RestrictDSClients	false	RTD	Set this to false if there is no concern about which hosts can send Decision Service Requests. If set to true (its default value), then DS will reject requests from hosts not in TrustedDSClients and not part of the Oracle RTD cluster.
TrustedDSClients	N/A	RTD	When RestrictDSClients is true (its default value), the set of hosts allowed to send Decision Service requests can be set here as a system property. However, it is easier to add new client hosts if TrustedDSClients is managed through JMX instead of as a system property. See Section 13.5.6, "Adding Trusted Decision Service Clients."

13.4.2.2 Generic JVM Options

This section describes generic JVM options and suggested values for the options.

- Set the option `-Djava.net.preferIPv4Stack=true`
- **Minimum Heap Size**

  Set this to `-Xms512M`.

  On development machines, this can be lower (for example, 128M) to reduce the required memory.
- **Maximum Heap Size**

  Set this to `-Xmx1024M`.

  On production machines, 2048M is a common value.
- **PermGen Size** *(not for JRockit JVM)*

  Set this to `-XX:PermSize=256m`.

  This is to reduce out-of-memory errors when redeploying Inline Services, because redeploying Inline Services temporarily introduces more java classes.
- **JIT** *(for Sun JVM)*

  Set this to `-server`.

13.4.2.3 Application Server Specific JVM Options

This section contains application server specific JVM options.

13.4.2.3.1 Oracle AS JVM Options

- **JConsole Setup**

  See Section 13.4.3.1, "JConsole on Oracle AS and WebLogic."

13.4.2.3.2 WebLogic JVM Options

- **JConsole Setup**

  See Section 13.4.3.1, "JConsole on Oracle AS and WebLogic."

- **EMF Registry**

  Set this system property to specify the EMF (Eclipse Metadata Framework) registry implementation used by Oracle RTD.

  Property Name: **'org.eclipse.emf.ecore.EPackage.Registry.INSTANCE'**

  Property Value: **'com.sigmadynamics.emf.util.SDEMFRegistry'**

13.4.2.3.3 WebSphere JVM Options

- **EMF Registry**

  Set this system property to specify the EMF (Eclipse Metadata Framework) registry implementation used by Oracle RTD.

  Property Name: **'org.eclipse.emf.ecore.EPackage.Registry.INSTANCE'**

  Property Value: **'com.sigmadynamics.emf.util.SDEMFRegistry'**

- **Disable MBean Routing**

  Property Name: **'com.ibm.websphere.mbeans.disableRouting'**

  Property Value: **'<on>OracleRTD:*</on>'**

13.4.3 Setting Up JConsole Access

If JConsole is to be used as the JMX console, as opposed to some other MBean browser, its configuration is application server dependent. The only consideration here that is specific to clustering, as opposed to the general information provided elsewhere in this guide, is the need to use a different jmx port setting for each AS instance.

13.4.3.1 JConsole on Oracle AS and WebLogic

Set the following JVM system properties, to ensure that the port is unique on the host:

- -Dcom.sun.management.jmxremote.authenticate=false
- -Dcom.sun.management.jmxremote.ssl=false
- -Dcom.sun.management.jmxremote=true
- -Dcom.sun.management.jmxremote.port=12345

13.4.3.2 JConsole on WebSphere

To set up JConsole for WebSphere, see Section 4.12, "Setting Up JConsole for WebSphere."

Create a jconsole-startup batch file for each Oracle RTD instance.

13.4.4 Setting Up JDBC Resources

Before deploying Oracle RTD, you must create a JDBC connection pool and data source for each AS instance. When Oracle RTD is deployed to these AS instances, it must be able to open JDBC connections to a common shared data source, accessed by the JNDI lookup name, **jdbc/SDDS**.

For detailed application server specific instructions about creating the JDBC resources, refer to Chapter 8, "Configuring Data Access for Oracle Real-Time Decisions."

For instructions about how to create JDBC resources to a database shared by all AS instances, refer to your application server documentation. Depending on the J2EE application server vendor, you typically can create the connection pool and data source just once, or perhaps once per J2EE cluster.

13.5 Deploying Oracle RTD into an Oracle RTD Type 3 Cluster

This section describes in general terms how to deploy Oracle RTD into a clustered environment where Decision Service runs in separate Oracle RTD instances from the instances running Decision Center and Learning Service. This is an Oracle RTD Type 3 cluster, and this is expected to be the most common configuration used in production.

Generalizing these instructions for other Oracle RTD cluster types is straightforward if the following rules are followed.

1. Deploy `RTD.ear` into separate J2EE clusters for every combination of Oracle RTD services that need to be co-located in the same Oracle RTD instance. This is illustrated in the table below.
2. In each deployment of `RTD.ear`, change the web context roots associated with the services that will not be used, effectively hiding them so the load balancer will not send requests to Oracle RTD instances for disabled services.
3. Disable DS and LS, through JConsole, in the Oracle RTD instances where you do not want them to run, to avoid loading these services into memory.

Note: Currently DC is unconditionally loaded into memory, but will not create any sessions because of step 2 above.

Expanding on steps 1 and 3, you will need from one to three J2EE clusters, as detailed in the following table:

Oracle RTD Cluster Type	J2EE Clusters Required	Oracle RTD Services Enabled	Oracle RTD Services Disabled by JConsole
Type 1	rtdGroup1	DS, DC, LS	none
Type 2	rtdGroup1	DS, DC	LS
	rtdGroup2	LS	DS
Type 3	rtdGroup1	DS	LS
	rtdGroup2	DC, LS	DS
Type 4	rtdGroup1	DS	L:S
	rtdGroup2	DC	DS, LS
	rtdGroup3	LS	DS

Expanding on step 2, the following table shows the web context roots needed by the three Oracle RTD services when enabled:

Oracle RTD Service	Context Roots
DS	/rtis
DC	/ui
	/soap
	/schema
LS	/ls

The disabled services should be "hidden" by appending a string such as "-hide<*n*>" to their context roots, where <*n*> should be 1, 2, or 3, corresponding to the J2EE cluster number.

The use of <*n*> is to ensure that you do not use the same context root twice. It is really only useful in a Type 4 cluster, as this is the only Oracle RTD cluster type where you would be disabling the same Oracle RTD service type in more than one J2EE cluster.

Specifically for an Oracle RTD Type 3 cluster, deploy `RTD.ear` twice, as follows:

- First, into the J2EE cluster rtdGroup1, as application OracleRTD
- Then, into rtdGroup2 as application OracleRTD2

The rest of this section contains the following topics:

- Section 13.5.1, "Deploying RTD.ear Into rtdGroup1 as OracleRTD"
- Section 13.5.2, "Deploying RTD.ear Into rtdGroup2 as OracleRTD2"
- Section 13.5.3, "Disabling Certain Oracle RTD Services"
- Section 13.5.4, "Setting Decision Service Address"
- Section 13.5.5, "Adding Restricted Cluster Members"
- Section 13.5.6, "Adding Trusted Decision Service Clients"

13.5.1 Deploying RTD.ear Into rtdGroup1 as OracleRTD

Some application server types, like Oracle AS, enable you to save a deployment plan file that remembers deployment-time choices, so that you do not have to manually specify them again in subsequent deployments.

When you deploy for the first time:

- Use OracleRTD as the application name.
- Adjust the proposed web context-roots by appending "-hide" to each of them except /rtis.

  For example: /ui-hide, /ls-hide, /soap-hide, /schema-hide, /rtis.
- Save the new deployment plan on the local machine to use in subsequent deployments, if your application server supports saving deployment plans.

> **Tip:** Use a mnemonic name for the deployment plan file. For example, when the deployment enables only the Decision Service, call the deployment plan file **dsPlan.dat**.

13.5.2 Deploying RTD.ear Into rtdGroup2 as OracleRTD2

The considerations and steps for the deployment into rtdGroup2 are the same as for deployment into rtdGroup1, except for the following:

- The web-app context bindings will be different
- The application name will be different (OracleRTD2 instead of OracleRTD)

When you deploy for the first time:

- Use OracleRTD2 as the application name.
- Adjust the proposed web context-roots by appending "-hide" to /rtis.

  For example: /ui, /ls, /soap, /schema, /rtis-hide.
- For Oracle AS, change the library path settings to refer to the new application name: `${oracle.j2ee.home}/OracleRTD2` instead of `${oracle.j2ee.home}/OracleRTD`. For details, see Section 13.5.2.1, "Configuring Classpath for OracleRTD2 in Oracle AS."
- Save the new deployment plan on the local machine to use in subsequent deployments, if your application server supports saving deployment plans.

  If you follow the mnemonic-name convention for the deployment file name, you could use **dcLsPlan.dat** for the file name, as this deployment enables Decision Center and Learning Service.

13.5.2.1 Configuring Classpath for OracleRTD2 in Oracle AS

For Oracle AS 3.3, the last screen of the deployment process in Enterprise Manager allows you to adjust the class path used by various libraries. You must adjust the classpath because, as packaged in `RTD.ear`, the path assumes the application is named OracleRTD, but in this deployment it is OracleRTD2.

Click the Configure Class Loading activity.

Then scroll down to the Configure Application Libraries section, and replace the three occurrences of OracleRTD with OracleRTD2.

13.5.3 Disabling Certain Oracle RTD Services

After Oracle RTD has been started in all AS instances, use JConsole or another MBean browser to disable Decision Service or Learning Service in certain Oracle RTD instances according to the cluster type.

> **Note:** This is not necessary, and in fact cannot be done, if the services were enabled or disabled via system properties, as described in Section 13.4.2, "Setting JVM Options."

This only needs to be done the first time, or if you are changing cluster types. It is not necessary if you are redeploying Oracle RTD into the same cluster type, because these settings are saved in the SDDS database shared by all the instances.

For each instance in the Oracle RTD cluster, perform the following steps:

1. Start JConsole and connect to the instance.
2. Disable LS.

If the instance is in the J2EE cluster rtdGroup1, disable Learning Service, because these instances run just Decision Service.

Set the **LearningServiceEnabled** attribute to **false** in the MBean, **OracleRTD > SDPropertyManager > Misc**.

3. Disable DS.

   If the instance is in the J2EE cluster rtdGroup2, disable Decision Service, because these instances run just Learning Service and Decision Center.

   Set the **DecisionServiceEnabled** attribute to **false** in the MBean, **OracleRTD > SDPropertyManager > Misc**.

13.5.4 Setting Decision Service Address

For Oracle RTD Type 3 and Type 4 clusters, where DC and DS are not co-located, use JConsole to set the cluster's Decision Service address, so that DC can send test events to DS from its Interactive Integration Map view.

This can be done once in any instance. The change will be propagated automatically to all instances.

Set the DecisionServiceAddress attribute to a value like **http://myHost.com:8080** in the MBean, **OracleRTD > SDClusterPropertyManager > Misc**.

Generally, the host and port will be that of the HTTP server serving as the load-balancer for the `RTD.ear` deployment where DS is enabled. For the Type 3 cluster described in this chapter, this is the HTTP server/port hosting the web app for OracleRTD, not the one for OracleRTD2 - usually these would be same.

13.5.5 Adding Restricted Cluster Members

When the set of Oracle RTD cluster members is restricted to a specified list of hosts - a security precaution enabled via the system property RestrictClusterMembers - then additional hosts are more easily added through the use of JConsole, rather than through the system property TrustedClusterMembers, as described in Section 13.4.2.1, "Oracle RTD Clustering System Properties."

If TrustedClusterMembers is managed as a system property, then all Oracle RTD instances must be restarted, after manually adjusting this system property value for all Oracle RTD instances to include the new host.

When managing TrustedClusterMembers with JConsole, then no instances have to be restarted, except possibly the new instance if it was already running when its address was added to TrustedClusterMembers.

Oracle recommends the following sequence:

1. Determine the host IP address of the new Oracle RTD Cluster member. Host names are not sufficient, you need the IP address.
2. Through a JConsole connection to any existing Oracle RTD instance, add the new host address to the propertyTrustedClusterMembers in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**. This change should be propagated to all existing instances, and should be effective immediately.
3. Add the new AS instance to the J2EE cluster, and deploy Oracle RTD to the J2EE cluster.

13.5.6 Adding Trusted Decision Service Clients

When the set of Decision Service client hosts is restricted to a specified list of hosts - a security precaution enabled via the system property RestrictDSClients - then additional client hosts are more easily added through the use of JConsole, rather than through the system property TrustedDSClients, described in Section 13.4.2.1, "Oracle RTD Clustering System Properties."

If TrustedDSClients is managed as a system property, then all Oracle RTD instances must be restarted, after manually adjusting this system property value for all Oracle RTD instances to include the new client host. When managing TrustedDSClients with JConsole, then no instances have to be restarted.

Oracle recommends the following sequence:

1. Determine the host IP address of the new Decision Service client. Host names are not sufficient, you need the IP address.
2. Through a JConsole connection to any existing Oracle RTD instance, add the new host address to the property TrustedDSClients in the MBean, **OracleRTD > SDClusterPropertyManager > Cluster**. This change should be propagated to all existing instances, and should be effectively immediately.

13.6 Optional Session Affinity Cookie

Cookie name: `ORA_RTD_DSServerID`

Path: `/rtis`

Description: The contents of the cookie are a single integer identifying the RTD server instance that is hosting the request's Inline Service session.

This HTTP cookie is disabled be default, but can be enabled by setting the following MBean attribute to true:

- OracleRTD > SDClusterPropertyManager > Cluster: GenerateDSServerIdCookie

Its use has not been thoroughly tested, but it is believed that in some clustered installations it could reduce the number of times that requests need to be forwarded from one Decision Service instance to another, possibly improving performance. This cookie is not used for Decision Center, only for Decision Service, the service that handles Inline Service requests.

It is only needed in installations that could be using multiple HTTP sessions to deliver Inline Service requests to the same Inline Service session.

Because most external load balancers are routinely configured to support HTTP session affinity, and Inline Service session affinity is needed that spans multiple HTTP sessions, this cookie would be set in each of the HTTP sessions so that requests from the multiple HTTP sessions would be sent by the load balancer to the same Decision Service instance.

The usefulness of this cookie depends on each HTTP session being used to access just one Inline Service session.

Another assumption is that the load balancer's session affinity algorithm can be configured to use this cookie, instead of JSESSIONID as most are routinely configured to do.

13.7 Setting up a WebLogic Cluster

This section consists of the following topics:

13.7.1 Cluster Machine Setup

This section describes the overall hardware and software setup of the WebLogic cluster, to be used as the reference configuration for all the subsequent WebLogic related subsections.

Cluster Example: 2 Machines, 6 AppServers

Machine #1: <-- Admin Server, 3 Server Instances, Apache Server

- Oracle RTD Software
- WebLogic Application Server
- Apache HTTP Server

Machine #2: <-- 3 Server Instances

- WebLogic Application Server

13.7.2 Installing Oracle RTD Software

You must install the Oracle RTD software on machine #1, as follows:

1. Install `rtd_3.0.0_WL_win.zip` (or `rtd_3.0.0_WL_unix.cpio` if on Linux/Unix) on machine #1 (with WebLogic Admin Server).

13.7.3 Initializing Oracle RTD Database

You must initialize the Oracle RTD database on machine #1, as follows:

1. On machine #1, run *RTD_HOME*`\scripts\SDDBTool.cmd`.

   For more information, see Section 2.2.4, "Initializing the Oracle RTD Database Using SDDBTool."

13.7.4 Installing WebLogic \Apache Server

The first step is to install the WebLogic Application Server on both machines. Then you must install the Apache HTTP Server on machine #1.

13.7.4.1 Installing WebLogic Application Server

On machine #1 and machine #2 in the cluster, install WebLogic Application Server.

> **Note:** The following is to ensure the WebLogic Plug-in is also installed.

Perform the following steps on both machines:

1. In **Choose Install Type**, select **Custom**, then click **Next**.
2. In **Choose Components**, check the **WebLogic Server: Server** & **Web Server Plug-Ins** checkboxes, and uncheck all the others.

   Click **Next**.
3. In **Optional Tools**, uncheck **Mercury profiling tools**, then click **Next**.
4. In **Choose Product Installation Directories**, accept the defaults, then click **Next**.
5. In **Install Windows Service**, accept the defaults, then click **Next**.
6. In **Choose Shortcut Installation**, accept the defaults, then click **Next**.
7. Wait until the installation finishes.
8. When the installation is complete, click Done.

13.7.4.2 Installing Apache HTTP Server

Perform the following step:

1. On machine #1 in the cluster, install Apache HTTP Server 2.2.6.

13.7.5 Setting Up WebLogic Domain and Node Managers

The first steps are to create the WebLogic domain and to copy the OracleRTD JDBC jar to the domain. Then you must start the Node Managers on both machines.

13.7.5.1 Creating WebLogic Domain

To create the WebLogic domain, perform the following steps:

1. On machine #1 (your Admin Server machine), click **Start > All Programs > Oracle WebLogic > WebLogic Server 11gR1> Tools**, run **Configuration Wizard**.
2. In the Welcome dialog, select the **Create a new WebLogic Domain** radio button.
3. Click **Next**.

4. In the Domain Source dialog, select **Generate a domain configured automatically to support the following products** radio button.
5. Click **Next**.
6. In the Configure Administration Username and Password dialog, enter admin username/password, for example weblogic/weblogic.
7. Click **Next**.
8. In the Configure Server Start Mode and JDK dialog, under WebLogic Domain Startup Mode, select the **Development Mode** radio button
9. In the Configure Server Start Mode and JDK dialog, under JDK Selection, select a JDK that is supported for your system configuration, for example, Sun, JRockit.
10. Click **Next**.
11. In the Customize Environment and Services Settings dialog, select **No**.

    Note: this will Create Clusters and Servers using Admin Console.
12. Click **Next**.
13. In the Create WebLogic Domain dialog, for Domain Name, enter `oraclertd_domain` and click **Create**.
14. Wait until the installation finishes.
15. Check the **Start Admin Server** checkbox.
16. Click **Done**.

> **Note:** To manually start Admin Server, run *`weblogic-install-dir`*`/user_projects/domains/oraclertd_domain/bin/startWebLogic.cmd`.
>
> To manually stop Admin Server, run *`weblogic-install-dir`*`/user_projects/domains/oraclertd_domain/bin/stopWebLogic.cmd`

13.7.5.2 Copying OracleRTD JDBC Jar to Domain

To copy the OracleRTD JDBC jar to the domain, perform the following steps:

1. On machine #1 (your Admin Server machine) in *RTD_HOME*`\lib\jdbc\`, copy one of the following jar sets to *`weblogic-install-dir`*`/user_projects/domains/oraclertd_domain/lib/`:
   - SQLServer: `sqljdbc.jar`
   - Oracle: `ojdbc14.jar`
   - DB2: `db2jcc.jar` & `db2jcc_license_cu.jar`

> **Note:** This step removes the need to copy jdbc jars to each server instance machine.

13.7.5.3 Starting All Machine Node Managers

To start the machine Node Managers, perform the following steps:

1. Go to machine #1.

2. Start Node Manager by running: `*weblogic-install-dir*\wlserver_10.3\server\bin\startNodeManager.cmd`.

   Example: `c:\Oracle\Middleware\wlserver_10.3\server\bin\startNodeManager.cmd`

3. Repeat steps 1-2 for machine #2 in the cluster.

13.7.6 Creating Oracle RTD Roles and Users

Before proceeding, create Oracle RTD roles and users in WebLogic. See Section 5.8, "Creating Oracle RTD Roles and Users" for more information.

13.7.7 Creating Machines, Clusters, and Servers

This process consists of creating the WebLogic machines and the WebLogic cluster, then creating the servers and setting the server JVM properties.

13.7.7.1 Creating WebLogic Machines

To create the WebLogic machines, perform the following steps:

1. Logon to the Admin Console (machine #1):

   `http://*admin-server-host-name*:7001/console`

2. In the tree to the left under Change Center, click **Lock & Edit**.
3. In the tree to the left under Domain Structure, expand **Environment** and click **Machines**.
4. Click **New**.
5. For **Name**, enter a unique machine name, for example, `Machine-1`.
6. Click **OK**.
7. Click the machine name.
8. Under the Configuration tab, click **Node Manager**.
9. For **Listen Address**, enter the host name of a clustered server, for example, `myHostName`.
10. Click **Save**.
11. Repeat steps 3-10 for each machine in the cluster including the Admin Server machine.
12. In the tree to the left under Change Center, click **Activate Changes**.

13.7.7.2 Creating WebLogic Cluster

The following describes the cluster setup on the two machines:

- RTDCluster

  (Machine #1 and Node #1)

  – RTDServer1a

  – RTDServer1b

  – RTDServer1c

  (Machine #2 and Node #2)

- RTDServer2a
- RTDServer2b
- RTDServer2c

To create the WebLogic cluster, perform the following steps:

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 **Clusters**.
3. Click **New**.
4. For **Name**, enter `RTDCluster`.
5. Click **OK**.
6. In the tree to the left under Change Center, click **Activate Changes**.

13.7.7.3 Creating Servers

To create the servers, perform the following steps:

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 New
4. For **Server Name**, enter `RTDServer1a`.
5. For **Server Listen Port**, enter `7002`.
6. Under **Should this server belong to a cluster?**, select **Yes, make this server a member of an existing cluster**.
7. For **Select a cluster**, select **RTDCluster**.
8. Click **Next**.
9. Click **Finish**.
10. Click the server name **RTDServer1a**.
11. Under Configuration tab General section, for Machine, select **Machine-1**.
12. Click **Save**.
13. Repeat steps 2-12 for the following remaining server/port/machine configurations:
    - server = RTDServer1b, port = 7003, machine = Machine-1
    - server = RTDServer1c, port = 7004, machine = Machine-1
    - server = RTDServer2a, port = 7005, machine = Machine-2
    - server = RTDServer2b, port = 7006, machine = Machine-2
    - server = RTDServer2c, port = 7007, machine = Machine-2
14. In the tree to the left under Change Center, click **Activate Changes**.

13.7.7.4 Setting Server JVM Properties

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

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 server name **RTDServer1a**.
4. Under the Configuration tab, Server Start section, for **Arguments**, enter the following as a single line with spaces between properties (replacing *jmx-remote-port* with `12302`, and replacing *learning* with `true` for RTDServer1a and `false` for all other servers):

   -
     ```
     -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigma
     dynamics.emf.util.SDEMFRegistry

      -DRestrictClusterMembers=false

      -DSDGroupName=RTDCluster

       -DLearningServiceEnabled=learning

       -Dcom.sun.management.jmxremote=true

        -Dcom.sun.management.jmxremote.port=jmx-remote-port

        -Dcom.sun.management.jmxremote.authenticate=false

         -Dcom.sun.management.jmxremote.ssl=false

         -Djava.net.preferIPv4Stack=true
     ```

   Example:

   -
     ```
     -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigma
     dynamics.emf.util.SDEMFRegistry -DSDGroupName=RTDCluster
     -DRestrictClusterMembers=false
     -DLearningServiceEnabled=true
     -Dcom.sun.management.jmxremote=true
     -Dcom.sun.management.jmxremote.port=12302
     -Dcom.sun.management.jmxremote.authenticate=false
     -Dcom.sun.management.jmxremote.ssl=false
     -Djava.net.preferIPv4Stack=true
     ```

5. Click **Save**.
6. Repeat steps 2-5 for the following remaining server/jmx-remote-port configurations:

   - server = RTDServer1b, jmx-remote-port = 12303
   - server = RTDServer1c, jmx-remote-port = 12304
   - server = RTDServer2a, jmx-remote-port = 12305
   - server = RTDServer2b, jmx-remote-port = 12306
   - server = RTDServer2c, jmx-remote-port = 12307

7. In the tree to the left under Change Center, click **Activate Changes**.

13.7.8 Creating JDBC Data Sources

Notes: 1. For general information about configuring WebLogic JDBC data sources, see *Oracle Fusion Middleware Configuring and Managing JDBC for Oracle WebLogic Server*.

2. If your WebLogic version is 11g 10.3.3+ and your Oracle RTD database is in an Oracle database server, see Section 13.7.8.1, "Creating an Oracle JDBC Provider for the Oracle RTD Database on WebLogic 11g 10.3.3+."

To create the JDBC data sources, 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 `RTD_DS`.
5. For **JNDI Name**, enter `SDDS`.
6. For **Database Type**, select **Other**.
7. For **Database Driver**, select **Other**.
8. Click **Next**.
9. Uncheck the **Supports Global Transactions** checkbox.
10. Click **Next**.

    Perform the next set of steps that is appropriate to the database that holds your data source:

    If your database is Oracle, continue at step 11.

    If your database is SQLServer, continue at step 23.

    If your database is DB2, continue at step 35.

    Oracle

11. For **Database Name**, enter your Oracle SID name, for example, `ORCL`.
12. For **Host Name**, enter your database server name, for example, `mydbservername`.
13. For **Port**, enter your database port number, for example, 1521.
14. For **Database User Name**, enter your Oracle user name, for example, `myusername`.
15. For **Password** and **Confirm Password**, enter your Oracle user password, for example, `SD`.
16. Click **Next**.
17. For **Driver Class Name**, enter `oracle.jdbc.pool.OracleDataSource`.
18. For **URL**, enter `jdbc:oracle:thin:@`*`db_host:db_port:sid`*.

    For example, `jdbc:oracle:thin:@mydbservername:1521:ORCL`

19. For **Properties**, enter `user=`*`db_user_name`*, for example, `user=myusername`.
20. For **Test Table Name**, enter `SDApps`.
21. Click **Test Configuration**.

22. Click **Next**.

    Continue at step 47.

    SQLServer

23. For **Database Name**, enter your SQLServer database name, for example, `rtd`.
24. For **Host Name**, enter your database server name, for example, `mydbservername`.
25. For **Port**, enter your database port number, for example, `1433`.
26. For **Database User Name**, enter your SQLServer user name, for example, `myusername`.
27. For **Password** and **Confirm Password**, enter your SQLServer user password, for example, `sd`.
28. Click **Next**.
29. For **Driver Class Name**, enter `com.microsoft.sqlserver.jdbc.SQLServerDriver`
30. For **URL**, enter `jdbc:sqlserver://`*`db_host:db_port`*

    For example, `jdbc:sqlserver://mydbservername:1433`

    > **Note:** If the data source is on a SQL Server named instance, specify the *`db_host`* parameter using the format *`host_name`*`\`*`instance_name`*.

31. For **Properties**, enter each property on a new line:
    - `user=`*`db_user_name`*, for example, `user=myusername`
    - `DatabaseName=`*`db_name`*, for example, `DatabaseName=rtd`
32. For **Test Table Name**, enter `SDApps`.
33. Click **Test Configuration.**
34. Click **Next**.

    Continue at step 47.

    DB2

35. For **Database Name**, enter your DB2 database name, for example, `rtd`.
36. For **Host Name**, enter your database server name, for example, `mydbservername`.
37. For **Port**, enter your database port number, for example, `50000`.
38. For **Database User Name**, enter your DB2 user name, for example, `myusername`.
39. For **Password** and **Confirm Password**, enter your DB2 user password, for example, `SD`.
40. Click **Next**.
41. For **Driver Class Name**, enter `com.ibm.db2.jcc.DB2Driver`.
42. For **URL**, enter `jdbc:db2://`*`db_host:db_port/db_name`*

    For example, `jdbc:db2://mydbservername:50000/rtd`

43. For **Properties**, enter `user=`*`db_user_name`*, for example, `user=myusername`.
44. For **Test Table Name**, enter `SDApps`.
45. Click **Test Configuration**.
46. Click **Next**.

    Continue at step 47.
47. Under Clusters, check the **RTDCluster** checkbox.
48. Click **Finish**.
49. In the tree to the left under Change Center, click **Activate Changes**.

13.7.8.1 Creating an Oracle JDBC Provider for the Oracle RTD Database on WebLogic 11g 10.3.3+

To create a JDBC provider when your Oracle RTD database is in an Oracle database server, and Oracle RTD is deployed on WebLogic 11g 10.3.3+, perform the following steps:

1. Log into the WebLogic Server Administration Console with the administrator user name and password.
2. Navigate the path **Services** -> **Data Sources** -> **New** -> **Generic Data Source**.
3. On the JDBC Data Source Properties page, follow these steps:
    a. For **Name**, enter a descriptive data source name, such as `RTD_DS`.
    b. For **JNDI Name**, enter `SDDS`.
    c. For **Database Type**, select **Oracle**.
    d. For the **Database Driver,** select **Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1 and later**, then click **Next**.
4. On the Transaction Options page, deselect **Supports Global Transactions**, then click **Next**.
5. On the Connection Properties page, follow these steps:
    a. For **Database Name**, enter the name of the Oracle RTD Database you created in Section 2.2.
    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**.
6. On the Test Database Connection page, leave all the settings already filled, except enter `SDAPPS` for **Test Table Name**.
7. Click **Test Configuration**. If the test fails, go back and check your settings. If the test succeeds, click **Next**.
8. Select the cluster where you want the data source to be made available (for example, **RTD_Cluster**). You must perform this step before completing the data source configuration.
9. Click **Finish**, then click **Activate Changes**.

13.7.9 Setting Up the Web Server

This process consists of enabling the WebLogic proxy plug-in in the servers, then setting up the Apache HTTP server.

13.7.9.1 Enabling WebLogic Proxy Plug-In in Servers

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

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 server name **RTDServer1a**.
4. Under Configuration tab General section, click **Advanced**.
5. Under Advanced, check **WebLogic Plug-In Enabled** checkbox.
6. Click **Save**.
7. Repeat steps 2-6 for each of the following remaining servers:
   - RTDServer1b
   - RTDServer1c
   - RTDServer2a
   - RTDServer2b
   - RTDServer2c
8. In the tree to the left under Change Center, click **Activate Changes**.

13.7.9.2 Setting Up Apache HTTP Server

To set up the Apache HTTP server, perform the following steps:

1. On machine #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
   ```

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

   For example: `ServerName myHostName.us.oracle.com:8080`

5. At the end of `httpd.conf`, add the following lines (replacing hostName* and 7002-7007 in the `WebLogicCluster` line with your servers and ports created in Section 13.7.7.3, "Creating Servers"):

```
LoadModule weblogic_module modules/mod_wl_22.so
<IfModule mod_weblogic.c>
    # RTDCluster_DsDcLs
    <LocationMatch "/(ui|schema|soap|rtis|ls)">
      SetHandler weblogic-handler
      WebLogicCluster hostName1:7002,hostName1:7003,hostName1:7004,
                      hostName2:7005,hostName2:7006,hostName2:7007
    </LocationMatch>
 </IfModule>
```

6. Save `httpd.conf`.

7. Start Apache HTTP: **Start>All Programs>Apache HTTP Server 2.2.6 >Control Apache Server>Start**

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

   > **Note:** To stop, run: **Start>All Programs>Apache HTTP Server 2.2.6 >Control Apache Server>Stop**
   >
   > (Or `C:\Program Files\Apache Software Foundation\Apache2.2\bin>httpd -k stop`)

13.7.10 Deploying RTD.ear Onto Cluster

To deploy RTD.ear onto the cluster, perform the following steps:

1. On machine #1, in *`RTD_HOME`*`/package`, unzip `RTD.ear` into a directory called **expanded**.
2. In the admin console, in the tree to the left under Change Center, click **Lock & Edit**.
3. In the tree to the left under Domain Structure, click **Deployments**.
4. Click **Install**.
5. For **Location**, find *`RTD_HOME`*`/package`.
6. Select **expanded**.
7. Click **Next**.
8. Select **Install this deployment as an application**.
9. Click **Next**.
10. Under Clusters, check the **RTDCluster** checkbox.
11. Click **Next**.
12. Under General, for **Name**, enter `OracleRTD`.
13. Under Source accessibility, select **Copy this application onto every target for me**.
14. Click **Next**.
15. Click **Finish**.
16. In the tree to the left under Change Center, click **Activate Changes**.

> **Note:** `RTD.ear` will be installed on every machine server instance in: *`weblogic-install-dir`*`\user_projects\domains\oraclertd_domain\servers\`*`server-instance-name`*`\stage\OracleRTD\expanded`

13.7.11 Copying WebLogic Domain to All Clustered Machines

To copy the WebLogic domain to all clustered machines, perform the following steps:

1. Go to machine #2 in the cluster.
2. Start the WebLogic Scripting Tool by running:

   ```
   weblogic-install-dir\wlserver_10.3\common\bin\wlst.cmd
   ```

   For example: `C:\bea\wlserver_10.3\common\bin\wlst.cmd`
3. Connect to machine #1 (Admin server) by entering:

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

   For example: wls:/offline>
   ```
   connect('weblogic','weblogic','t3://myHostName.us.oracle.com:
   7001')
   ```
4. Enroll the machine as a managed server by entering:

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

   For example: wls:/oraclertd_domain/serverConfig>

   ```
   nmEnroll('C:/bea/user_projects/domains/oraclertd_domain',
   'C:/bea/wlserver_10.3/common/nodemanager')
   ```
5. Run `disconnect()`.

   For example: wls:/oraclertd_domain/serverConfig> `disconnect()`
6. Run `exit()`.

   For example: wls:/offline> `exit()`.

13.7.12 Starting the Cluster

This process consists of restarting the Admin Server and Node Managers, then starting the clustered servers.

13.7.12.1 Restarting Admin Server and Node Managers

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

1. On machine #1, find the command window where the Node Manager has been started (from startNodeManager.cmd) and hit Ctrl-C to stop it.
2. Repeat step 1 for machine #2.
3. On machine #1 (Admin Server machine), run:

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

   For example:

```
c:/Oracle/Middleware/user_projects/domains
/oraclertd_domain/bin/stopWebLogic.cmd
```

4. On machine #1 (Admin Server machine), run:

   weblogic-install-dir/user_projects/domains/oraclertd_domain/bin/startWebLogic.cmd

   For example:

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

5. On machine #1 (Admin Server machine), run

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

6. Repeat step 5 for machine #2.

13.7.12.2 Starting Clustered Servers

To start the clustered servers, perform the following steps:

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

Logs for Machine 1:

- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer1a\logs\RTDServer1a.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer1b\logs\RTDServer1b.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer1c\logs\RTDServer1c.log`

Logs for Machine 2:

- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer2a\logs\RTDServer2a.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer2b\logs\RTDServer2b.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer2c\logs\RTDServer2c.log`

13.7.13 Starting the OracleRTD Application

To start the OracleRTD application, perform the following steps:

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.

Logs for Machine 1:

- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer1a\stage\OracleRTD\expanded\log\server.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer1b\stage\OracleRTD\expanded\log\server.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer1c\stage\OracleRTD\expanded\log\server.log`

Logs for Machine 2:

- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer2a\stage\OracleRTD\expanded\log\server.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer2b\stage\OracleRTD\expanded\log\server.log`
- `C:\Oracle\Middleware\user_projects\domains\oraclertd_domain\servers\RTDServer2c\stage\OracleRTD\expanded\log\server.log`

13.7.14 Setting JConsole Server Properties

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

1. On machine #1 or machine #2, run `jconsole.exe` from a Java 1.5 installation.

   For example: `C:\Program Files\Java\jdk1.5.0_10\bin\jconsole.exe`
2. In JConsole, connect to the Agent dialog, then click the Remote tab.
3. For **Host** or **IP**, enter the machine name of the first clustered server instance.

   For example: `myHostName1.us.oracle.com`
4. For **Port**, enter the JMX remote port set for this server instance in Section 13.7.7.3, "Creating Servers."

   For example: `12302`
5. Click **Connect**.
6. Set any other JConsole properties as you require.

13.7.15 Disabling HTTP Session Affinity and Enable Request Forwarding on Each Server

To disable HTTP session affinity and to enable request forwarding on each server, perform the following steps:

1. On machine #1 or machine #2, run `jconsole.exe` from a Java 1.6 installation.

   For example: `C:\Program Files\Java\jdk1.6.0_12\bin\jconsole.exe`
2. In JConsole, connect to the Agent dialog, then click the Remote tab.
3. For **Host** or **IP**, enter the machine name of the first clustered server instance.

   For example: `myHostName1.us.oracle.com`

4. For **Port**, enter the JMX remote port set for this server instance in Section 13.7.7.3, "Creating Servers."

   For example: `12302`

5. Click **Connect**.
6. In the tree to the left, expand **OracleRTD > SDClusterPropertyManager > Cluster**.
7. For **GenerateDSCookies**, enter `false`. Hit the Enter key.
8. In the tree to the left, expand **OracleRTD > SDClusterPropertyManager > Misc**.
9. For **DSManagesSessionAffinity**, enter `true`. Hit the Enter key.

   > **Note:** Disable request forwarding by setting **DSManagesSessionAffinity** to **false**.

10. For **DSEnforcesStrictSessionAffinity**, enter `true`. Hit the Enter key.

13.7.16 Deploying CrossSell Inline Service (Through Web Server Port 8080)

To deploy the CrossSell Inline Service through web server port 8080, perform the following steps:

1. On machine #1, start Oracle RTD Studio by running *`RTD_HOME`*`\eclipse\eclipse.exe`.
2. In top menu bar, click **File** then **Import**.
3. Select **Existing Projects into Workspace**.
4. Click **Next**.
5. Click **Browse...**
6. Go to *`RTD_HOME`*`\examples\CrossSell`.
7. Click **OK**.
8. Click **Finish**.
9. Wait until the Inline Service build is finished.
10. In top menu bar, click the icon **Deploy the Inline Service to a server**.
11. Click **Deploy**.

    > **Note:** By default, Studio will deploy to **localhost:8080** (which is machine #1 and the web server "load balancer" port)

12. For **User Name**, enter `sdsu`.
13. Click **Connect**.
14. Wait until the CrossSell Inline Service is deployed.

13.7.17 Creating CrossSell Database Tables

To create the CrossSell database tables, perform the following steps:

1. On machine #1, in a command console:
   cd to *RTD_HOME*\examples\CrossSell\etc\data\Oracle
2. Run the following command:

   ```
   InitAppDB.cmd RTD_HOME db_host_name port sid db_user_name
   admin_user_name admin_user_password
   ```

   For example:

   ```
   InitAppDB.cmd C:\OracleBI\RTD mydbmachine.us.oracle.com 1521
   orcl MYUSERNAME MYADMINUSERNAME MYADMINPASSWORD
   ```

> **Note:** If the database is on a SQL Server named instance, specify the *db_host_name* parameter using the format *host_name\instance_name*.

13.7.18 Running Load Generator Against CrossSell Inline Service (Through Web Server Port 8080)

To run Load Generator against the CrossSell Inline Service through web server port 8080, perform the following steps:

1. On machine #1, start Oracle RTD Load Generator by running:

   ```
   RTD_HOME\scripts\loadgen.cmd
   ```

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

   Select **clientHttpEndPoints.properties** (which is already set to localhost:8080 or machine #1).
8. Click **Open**.
9. In the top menu bar, click **File,** then **Save**.
10. In the top menu bar, click the icon **Runs the current Load Generator script** (blue right arrow).

13.8 Setting up a WebSphere Cluster

Oracle RTD is supported on both UNIX and Windows platforms for IBM WebSphere application server.

This section contains the following topics:

- Section 13.8.1, "Cluster Machine Setup"
- Section 13.8.2, "Setting Up Host Machine Files"
- Section 13.8.3, "Installing WebSphere Software"
- Section 13.8.4, "Installing Oracle RTD Software"

- Section 13.8.5, "Initializing the Oracle RTD Database"
- Section 13.8.6, "Adding Nodes to Deployment Manager"
- Section 13.8.7, "Creating an Administrative User and Enabling Security"
- Section 13.8.8, "Adding Virtual Host Port 8080"
- Section 13.8.9, "Creating Cluster and Servers and Setting Server Properties"
- Section 13.8.10, "Creating Oracle RTD Roles and Users"
- Section 13.8.11, "Adding Virtual Host HTTP Address for Servers"
- Section 13.8.12, "Creating JDBC Providers and Data Sources"
- Section 13.8.13, "Creating the Web Server"
- Section 13.8.14, "Starting the Cluster and Servers"
- Section 13.8.15, "Deploying RTD.ear onto Cluster"
- Section 13.8.16, "Setting Classloader Priority"
- Section 13.8.17, "Viewing and Changing User-Role Associations"
- Section 13.8.18, "Creating Custom Roles and Assigning Permissions to Custom Roles (Optional)"
- Section 13.8.19, "Creating startJConsole Batch Files"
- Section 13.8.20, "Deploying CrossSell Inline Service"
- Section 13.8.21, "Creating CrossSell Database Tables"
- Section 13.8.22, "Running Load Generator Against CrossSell Inline Service"
- Section 13.8.23, "Creating WebSphere Thread Dump"

13.8.1 Cluster Machine Setup

This section describes the overall hardware and software setup of the WebSphere cluster, to be used as the reference configuration for all the subsequent WebSphere related subsections.

Cluster Example: 2 Machines, 6 AppServers

Machine #1: <-- Admin Server, IBM HTTP Server, 3 Server Instances (vms)

- Oracle RTD Software
- IBM Update Installer and FixPacks
- WebSphere Network Development
- IBM HTTP Server
- WebSphere Application Server

Machine #2: <-- 3 Server Instances (vms)

- Oracle RTD JDBC Driver Jar(s)
- IBM Update Installer and FixPacks
- WebSphere Application Server

13.8.2 Setting Up Host Machine Files

To set up the host machine files, perform the following steps:

1. On machine #1 in the cluster, open `C:\WINDOWS\system32\drivers\etc\hosts`.
2. Add the other machine IP address and fully qualified name participating in the cluster.

   For example:

   - `10.138.234.116 mymachine.us.oracle.com`
3. Repeat steps 1-2 for machine #2 in the cluster.

13.8.3 Installing WebSphere Software

Refer to the WebSphere documentation for information about installing WebSphere. Install the IBM updater and any download FixPacks, the WebSphere application server, WebSphere Network Deployment, then the IBM HTTP Server.

Note that Core and SDK fix packs are required for cluster setup. The basic outline procedure is:

- On both machines in the cluster, install Update Installer, and download any required fix packs for Core and SDK.
- On both machines in the cluster, install WebSphere Application Server, then use Update Installer to apply fix packs for Core and SDK.
- On machine #1, install Network Deployment, then use Update Installer to apply fix packs for Core and SDK.
- On machine #1, install HttpServer, then use Update Installer to apply fix packs for Core and SDK.

For details about the versions supported, see *System Requirements and Supported Platforms for Oracle Real-Time Decisions*.

13.8.4 Installing Oracle RTD Software

To install the Oracle RTD software, perform the following steps:

1. Install `rtd_websphere_3.0.0.zip` for WebSphere on machine #1 (with WebSphere Network Deployment).
2. Copy the RTD jdbc jar(s) to machine #2 in the cluster.

   Use the same directory structure *RTD_HOME*`\lib\jdbc\` as Machine #1:

   - SQLServer: *RTD_HOME*`\lib\jdbc\sqljdbc.jar`
   - Oracle: *RTD_HOME*`\lib\jdbc\ojdbc14.jar`
   - DB2: *RTD_HOME*`\lib\jdbc\db2jcc.jar`

     RTD_HOME`\lib\jdbc\db2jcc_license_cu.jar`

13.8.5 Initializing the Oracle RTD Database

To initialize the Oracle RTD database, perform the following step:

1. On machine #1, run *RTD_HOME*`\scripts\SDDBTool.cmd`

   For more information, see Section 2.2.4, "Initializing the Oracle RTD Database Using SDDBTool" in *Oracle Real-Time Decisions Installation and Administration Guide*.

13.8.6 Adding Nodes to Deployment Manager

Start the Deployment Manager, then, on both machines, add two nodes to the Deployment Manager.

The instructions in this section provide general information about adding nodes - for more specific information, refer to the WebSphere documentation as required.

The basic outline procedure for both machines in the cluster is:

- Start the Deployment Manager.

  For example, from the WebSphere Network Deployment Install directory, run the `bin\StartManager.bat` script.

- Add a node.

  For example, from the WebSphere Server Install directory, run the `bin\addNode.bat` script, specifying the network deployment machine name for the node.

  This adds Node Agent to Deployment Manager and also starts the node agent.

13.8.7 Creating an Administrative User and Enabling Security

If you have already enabled security in WebSphere, you can skip this section. You should still check that Java 2 security is not turned on.

Use the WebSphere Application Server Network Deployment Manager administrative console, called the Integrated Solutions Console, to create an administrative user and to enable security in WebSphere. For more information about how to use the Integrated Solutions Console, refer to the WebSphere documentation.

If security is disabled, follow these steps to enable security in WebSphere:

1. Access the Integrated Solutions Console at the URL `http://`*`websphere_host`*`:`*`port`*`/ibm/console`. At the login prompt, enter any user name. You will not need to enter any password. On Windows, you can also access the Integrated Solutions Console through **Start > Programs**.

   If you do not know the port number for the Integrated Solutions Console, you can find it in the `virtualhosts.xml` file, located in *`WEBSPHERE_HOME`*`/AppServer/profiles/`*`profile_name`*`/config/cells/`*`host_name`*.

2. In the tree on the left, expand **Security**, and choose **Secure administration, applications, and infrastructure**.

3. In the User account repository area, under the Available realm definitions heading, select **Federated repositories**, then click **Set as current**.

4. Click **Apply**, then **Save**.

5. Click **Configure**.

6. In the Federated repositories window, in the General Properties area, perform the following:

   a. Enter a **Realm Name**, such as RTDRealm.

   b. Enter a **Primary administrative user name**, such as admin.

   c. Under the Server user identity heading, select **Automatically generated server identity**.

7. In the Administrative user password window, in the General Properties area, enter the administrative user password in both the **Password** and **Confirm password** fields.
8. Click **OK**, then **Save**.
9. Log out, stop, then restart WebSphere.
10. Log in to the Integrated Solutions Console.

    If you do not know the port number for the Integrated Solutions Console, you can find it in the `virtualhosts.xml` file, located in *`WEBSPHERE_HOME`*`/AppServer/profiles/`*`profile_name`*`/config/cells/`*`host_name`*.
11. In the tree on the left, expand **Security** and choose **Secure administration, applications, and infrastructure**.
12. Under the Administrative security heading, select **Enable administrative security**.
13. Under the Application security heading, select **Enable application security**.
14. Under the Java 2 security heading, uncheck **Use Java 2 security to restrict application access to local resources**.
15. Click **Apply**, then **Save**.
16. Log out, stop, then restart WebSphere.

13.8.8 Adding Virtual Host Port 8080

To add virtual host port 8080, perform the following steps:

1. In the tree to the left, expand **Environment** and click **Virtual Hosts**.
2. Click **default host**.
3. Under Additional Properties, click **Host Aliases.**
4. Click **New**.
5. For Host Name, enter `'*'`.
6. For Port, enter `8080`.
7. Click **OK**.
8. Under Messages, click **Preferences** and check the **Synchronize changes with Nodes** checkbox.

    This ensures that future changes will be propagated to all nodes in cluster, provided that node agents are running.
9. Click **Apply**.
10. Click **Save**.

13.8.9 Creating Cluster and Servers and Setting Server Properties

This process consists of creating the cluster and servers, then setting the server JVM and administration properties.

The following describes the cluster setup on the two machines:

- RTDCluster

  (Machine #1 and Node #1)

  - RTDServer1a (LearningService = enabled)

- RTDServer1b
- RTDServer1c

(Machine #2 and Node #2)

- RTDServer2a
- RTDServer2b
- RTDServer2c

13.8.9.1 Creating Cluster and Servers

To create the cluster and servers, perform the following steps:

1. In the tree to the left, expand **Servers** and select **Clusters**.
2. Click **New**.
3. Step 1: For **Cluster name**, enter `RTDCluster`.
4. Step 1: Click **Next**.
5. Step 2: For **Member name**, enter `RTDServer1a`.

   Step 2: For **Select node**, select the first node created in Section 13.8.6, "Adding Nodes to Deployment Manager."
6. Step 2: Click **Next**.
7. Step 3: For **Member name**, enter `RTDServer1b`.

   Step 3: For **Select node**, select the same node (Node 1).
8. Step 3: Click **Add Member**.
9. Step 3: For **Member name**, enter `RTDServer1c`.

   Step 3: For **Select node**, select the same node (Node 1).
10. Step 3: Click **Add Member**.
11. Step 3: For **Member name**, enter `RTDServer2a`.

    Step 3: For **Select node**, select the second node created in Section 13.8.6, "Adding Nodes to Deployment Manager."
12. Step 3: Click **Add Member**.
13. Step 3: For **Member name**, enter `RTDServer2b`.

    Step 3: For **Select node**, select the same node (Node 2).
14. Step 3: Click Add Member.
15. Step 3: For **Member name**, enter `RTDServer2c`.

    Step 3: For **Select node**, select the same node (Node 2).
16. Step 3: Click **Add Member**.
17. Step 3: Click **Next**.
18. Step 4: Click **Finish**.
19. Under Messages, click **Save**.

13.8.9.2 Setting Server JVM Properties

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

1. In the tree to the left, expand **Servers** and select **Application servers**.
2. Click **RTDServer1a**.
3. Under Server Infrastructure, expand **Java and Process Management**.
4. Click **Process Definition.**
5. Under Additional Properties, click **Java Virtual Machine**.
6. Under the General Properties heading, in the **Generic JVM arguments** field, add the following string:

   ```
   -Djava.net.preferIPv4Stack=true
   ```

   If there is already a value in this field, add a space after the existing value, then add the new string.
7. Click **OK**.

   When you click **OK**, you may see an error stating that you need to provide values for **Initial Heap Size** and **Maximum Heap Size**. If you see this error, set these values as needed for your system (for example, you can set **Initial Heap Size** to `512` and **Maximum Heap Size** to `2048`).
8. Under Additional Properties, click **Custom Properties**.
9. Click **New**.
10. For **Name**, enter `org.eclipse.emf.ecore.EPackage.Registry.INSTANCE`
11. For **Value**, enter `com.sigmadynamics.emf.util.SDEMFRegistry`
12. Click **New**.
13. For **Name**, enter `rtd.instanceName`.
14. For **Value**, enter `RTDServer1a` (or `RTDServer1b`, `RTDServer1c`, `RTDServer2a`, `RTDServer2b`, `RTDServer2c`).
15. Click **New**.
16. For **Name**, enter `rtd.instancesShareRuntimeHome.`
17. For **Value**, enter `true`.
18. Click **New**.
19. For **Name**, enter `RestrictClusterMembers`.
20. For **Value**, enter `false`.
21. Click **New**.
22. For **Name**, enter `SDGroupName`.
23. For **Value**, enter `RTDCluster.`
24. Click **New**.
25. For **Name**, enter `LearningServiceEnabled`.
26. For **Value**, if server is RTDServer1a, enter `true`, otherwise enter `false`.
27. Under Messages, click **Save**.
28. Repeat step 1, then steps 3-27 for servers **RTDServer1b**, **RTDServer1c**, **RTDServer2a**, **RTDServer2b**, **RTDServer2c**.

13.8.9.3 Setting Server Administration Properties

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

1. In the tree to the left, expand **Servers**, and select **Application servers**.
2. Click **RTDServer1a**.
3. Under Server Infrastructure, expand **Administration**, and select **Administration Services**.
4. Under Additional Properties, click **Custom Properties**.
5. Click **New**.
6. For **Name**, enter `com.ibm.websphere.mbeans.disableRouting`.
7. For **Value**, enter `<on>OracleRTD:*</on>`.
8. Repeat step 1, then steps 3-7 for servers **RTDServer1b**, **RTDServer1c**, **RTDServer2a**, **RTDServer2b**, **RTDServer2c**.
9. Under Messages, click **Save**, then click **OK**.

13.8.10 Creating Oracle RTD Roles and Users

This section consists of the following topics:

- Section 13.8.10.1, "Creating Users for Oracle RTD"
- Section 13.8.10.2, "Creating Groups"
- Section 13.8.10.3, "Standard Oracle RTD Roles"

13.8.10.1 Creating Users for Oracle RTD

To create users, perform the following steps:

1. Log into the Integrated Solutions Console, using the administrative user and password.
2. In the tree on the left, expand **Users and Groups**, and choose **Manage Users.**
3. Click the **Create...** button.
4. Enter user and password information for the user, such as rtdadmin for the User ID.

   You will use the User ID later on, when you add the user to one or more groups.
5. Click **Create**, then **Close**.
6. Repeat steps 3 to 5 to create other users.

13.8.10.2 Creating Groups

To create groups, perform the following steps:

1. Log into the Integrated Solutions Console, using the administrative user and password.
2. In the tree on the left, under **Users and Groups**, choose **Manage Groups.**
3. Click the **Create...** button.
4. In the Group name field, enter RTDAdminGroup.
5. Click **Create**, then **Close**.

6. In the Manage Groups page, click the group name RTDAdminGroup.
7. Click the **Add Users...** button.
8. Click **Search** to display a list of users.
9. In the search result list, select the user name to add to the RTDAdminGroup.
10. Click **Add**, then **Close**.
11. Repeat steps 2 through 10 to create each of the following groups for Oracle RTD:
    - RTDDCEditorGroup
    - RTDDCUserGroup
    - RTDStudioDeployerGroup
    - RTDStudioDownloaderGroup
    - RTDBatchAdminGroup
    - RTDChoiceEditorGroup
    - RTDUserGroup
12. In the Manage Groups page, click the group name RTDUserGroup.
13. In the Group Properties area, click the Members tab.
14. Click the **Add Groups...** button.
15. Add all the groups created for Oracle RTD except RTDUserGroup.
16. Click **Close**.

13.8.10.3 Standard Oracle RTD Roles

The groups specified in Section 13.8.10.2, "Creating Groups" are automatically mapped to the standard Oracle RTD roles, as shown in Table 13–5.

Table 13–5 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

Section 7.2, "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.

> **Note:** For information about custom roles, see Section 13.8.18, "Creating Custom Roles and Assigning Permissions to Custom Roles (Optional)."

13.8.11 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.

13.8.11.1 Getting HTTP Addresses For Each Server

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

1. In the tree to the left, expand **Servers** and select **Application Servers**.
2. Click **RTDServer1a.**
3. Under Communications, expand **Ports**.
4. Write down the *`WC_defaulthost`* for this server, for example: 9081.

   This is used to connect with this server via HTTP.
5. Repeat steps 1-4 for servers `RTDServer1b`, `RTDServer1c`, `RTDServer2a`, `RTDServer2b`, `RTDServer2c`.

13.8.11.2 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:

1. In the tree to the left, expand **Environment** and click **Virtual Hosts**.
2. Click **default host**.
3. Under Additional Properties, click **Host Aliases**.
4. Click **New**.
5. For **Host Name**, enter `'*'`.
6. For **Port**, enter the *`WC_defaulthost`* port for server RTDServer1a, for example, `9081`.

   Note: You do not have to add the same port a second time for different machines.
7. Click **OK**.
8. Click **Save**.
9. Repeat steps 1-8 for each unique remaining server port, for example: `9082`, `9083`.

13.8.12 Creating JDBC Providers and Data Sources

This process consists of the following:

- Creating JDBC Providers
- Creating J2C Authentication Aliases
- Creating Data Sources
- Setting Data Source Properties
- Setting Data Source Statement Cache
- Restarting Nodes and Deployment Manager
- Testing Data Source Connection

13.8.12.1 Creating JDBC Providers

To create JDBC providers, perform the following steps:

1. In the tree to the left, expand **Resources**, expand **JDBC**, and select **JDBC Providers.**
2. Under Scope, select the cell scope, for example, **Cell=myMachineCell01**.
3. Click **New**.
4. Step 1: For **Database type**, select **User-defined**.

   Perform the steps appropriate to your data source.

   If your data source is SQLServer, continue at step 5.

   If your data source is Oracle, continue at step 9.

   If your data source is DB2, continue at step 13.

   SQLServer
5. Step 1: For **Information required Implementation class name**,

   enter `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`
6. Step 1: For **Name**, enter `RTDDataProvider.`
7. Step 1: Click **Next**.
8. Step 2: For **Class path**, enter `*RTD_HOME*/lib/jdbc/sqljdbc.jar`.

   For example, `C:/OracleBI/RTD/lib/jdbc/sqljdbc.jar`

   > **Note:** The path must be the same for each machine.

   Continue at step 17.

   Oracle
9. Step 1: For **Information required Implementation class name**,

   enter `oracle.jdbc.pool.OracleConnectionPoolDataSource`
10. Step 1: For **Name**, enter `RTDDataProvider`.
11. Step 1: Click **Next**.
12. Step 2: For **Class path**: enter `*RTD_HOME*/lib/jdbc/ojdbc14.jar`

    For example, `C:/OracleBI/RTD/lib/jdbc/ojdbc14.jar`

    Continue at step 17.

    DB2
13. Step 1: **For Information required Implementation class name**,

    enter `com.ibm.db2.jcc.DB2ConnectionPoolDataSource`
14. Step 1: For **Name**, enter `RTDDataProvider.`
15. Step 1: Click **Next**.
16. Step 2: For **Class path:** enter `*RTD_HOME*/lib/jdbc/db2jcc.jar;*RTD_HOME*/lib/jdbc/db2jcc_license_cu.jar`

For example:
`C:/OracleBI/RTD/lib/jdbc/db2jcc.jar;C:/OracleBI/RTD/lib/jdbc/db2jcc_license_cu.jar`

Note: The path must be the same for each machine.

Continue at step 17.

17. Step 2: Click **Next**.
18. Step 3: Click **Finish**.
19. Under Messages, click **Save**.

13.8.12.2 Creating J2C Authentication Aliases

To create J2C authentication aliases, perform the following steps:

1. In the tree to the left, expand **Security**, and select **Secure administration, applications, and infrastructure**.
2. Expand **Java Authentication and Authorization Service** and select **J2C authentication data**.
3. Click **New**.
4. For **Alias**, enter `RTDDS_auth`.
5. For **User ID**, enter the name of the database run-time user, for example, `sa1`.
6. For **Password**, enter the corresponding password for the database user, for example, `sa1`.
7. Click **OK**.
8. Under Messages, click **Save**.

   The J2C alias will be created in cell manager scope (host_nameCell_manager_name).

13.8.12.3 Creating Data Sources

To create data sources, perform the following steps:

1. In the tree to the left, expand **Resources**, expand **JDBC**, and select **JDBC Providers**.
2. Click **RTDDataProvider**.
3. Under Additional Properties, click **Data Sources**.
4. Select the Cluster scope, then click **New**.
5. Step 1: For Data source name, enter `RTD_DS` .
6. Step 1: For JNDI name, enter `SDDS`.
7. Step 1: Under **Component-managed authentication alias and XA recovery authentication alias**, select **RTDDS_auth**.
8. Step 1: Click **Next**.
9. Step 2: Click **Next**.
10. Step 3: Click **Finish**.

11. Under Messages, click **Save.**

13.8.12.4 Setting Data Source Properties

To set data source properties, perform the following steps:

1. In the tree to the left, expand **Resources**, expand **JDBC**, and select Data Sources.
2. Click **RTD_DS**.

   Perform the steps appropriate to your data source.

   If your data source is SQLServer, continue at step 3.

   If your data source is Oracle, continue at step 6.

   If your data source is DB2, continue at step 9.

   SQLServer

3. Under Additional Properties, click **Custom properties**.
4. Click the **Select All** icon.
5. Enter your `database name`, `server name`, and `port number` into the fields provided.

   > **Note:** If the data source is on a SQL Server named instance, specify the database server name using the format *`host_name\instance_name`*.

   Oracle

6. Under **Oracle datasource properties**,

   for **URL**, enter `jdbc:oracle:thin:@`*`db_host:db_port:sid`*.

   For example, `jdbc:oracle:thin:@mydbhost:1521:orcl`.

7. Click **OK**.
8. Under Messages, click **Save** to end the process.

   DB2

9. Under **DB2 Universal data source properties**,

   for **Database name**, enter the name of your database, for example, `rtd`.

10. For **Driver type**, enter `4`.
11. For **Server name**, enter your database server name, for example, `localhost`.
12. For **Port number**, enter your database port number, for example, `60000`.
13. Click **OK**.
14. Under Messages, click **Save** to end the process.

13.8.12.5 Setting Data Source Statement Cache

To set data source statement cache, perform the following steps:

1. In the tree to the left, expand **Resources**, expand **JDBC**, and select **JDBC Providers**.
2. Click **RTDDataProvider**.

3. Under Additional Properties, click **Data Sources**.
4. Click **RTD_DS**.
5. Under Additional Properties, click **WebSphere Application Server data source properties**.
6. For **Statement cache size**, enter `0` (instead of 10).
7. Click **OK**.
8. Under Messages, click **Save**.

13.8.12.6 Restarting Nodes and Deployment Manager

This section describes how to restart the nodes and the Deployment manager

In the WebSphere administrative console, restart the nodes for each machine from System Administration > Node agents.

To restart the Deployment Manager, perform the following steps:

1. On machine #1, stop the Deployment Manager:

   ```
   websphere-network-deployment-install-dir\bin\stopManager.bat
   ```

   For example: `C:\Program Files\IBM\WebSphere\NetworkDeployment\bin\stopManager.bat`

2. On machine #1, start the Deployment Manager:

   ```
   websphere-network-deployment-install-dir\bin\startManager.bat
   ```

   For example: `C:\Program Files\IBM\WebSphere\NetworkDeployment\bin\startManager.bat`

13.8.12.7 Testing Data Source Connection

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

1. Log on to the Deployment Manager Admin Console:

   ```
   http://deployment-manager-host-name:9061/ibm/console
   ```

2. In the tree to the left, expand **Resources**, expand **JDBC**, and select **JDBC Providers**.
3. Click **RTDDataProvider**.
4. Under Additional Properties, click **Data Sources**.
5. Check the **RTD_DS** checkbox.
6. Click **Test connection**.

> **Note:** Deployment manager on machine #1 must be running for the Oracle RTD data source to be active (cell scope).

13.8.13 Creating the Web Server

This process consists of modifying the IBM HTTP server configuration file, creating the web server, generating and propagating the web server plug-in, then starting the web server.

13.8.13.1 Modifying IBM HTTP Server Configuration File

To modify the IBM HTTP server configuration file, perform the following steps:

1. On machine #1, open *`ibm-http-server-install-dir`*`\conf\httpd.conf`

   For example: `C:\Program Files\IBM\HTTPServer\conf\httpd.conf`

2. Change the port for property **AfpaPort** to `8080`.

   For example: `AfpaPort 8080`

3. Change the port for property **Listen** to `8080`.

   For example: `Listen 0.0.0.0:8080`

4. Change port for property **ServerName** to `8080`.

   For example: `ServerName localhost:8080`

5. Change **webserver1** to **RTDWebServer** for property **WebSpherePluginConfig**.

   > **Note:** You can leave the name as webserver1. If you do, then make sure that you use the name webserver1 instead of RTDWebServer for the rest of the instructions.

6. Save `httpd.conf`.

13.8.13.2 Creating Web Server

To create the web server, perform the following steps:

1. Back in the admin console, in the tree to the left, expand **Servers** and select **Web servers**.
2. Click **New**.
3. Step 1: For **Select node**, select the node where web server will reside, for example, *`host_nameNode_name`*. (Machine #1).
4. Step 1: For **Server name**, enter the name you used in the IBM HTTP `httpd.conf` file, for example, **RTDWebServer**.
5. Step 1: For **Type**, select **IBM HTTP Server**.
6. Step 1: Click **Next**.
7. Step 2: Click **Next**.
8. Step 3: For **Port** enter `8080`.
9. Step 3: For **Web server installation location,** enter *`ibm-http-server-install-dir`*.

   For example, `C:\Progra~1\IBM\HTTPServer`, not `C:\Progam Files\IBM\HTTPServer`

   Do not enter spaces for this directory.

10. For **Service name**, enter `IBMHTTPServer6.1`.
11. Step 3: For **Plug-in installation location,** enter *`ibm-http-server-install-dir`*`\Plugins`

    For example, `C:\Program Files\IBM\HTTPServer\Plugins`

12. Step 3: Click **Next**.

13. Step 4: Click **Finish**.

14. Under Messages, click **Save.**

13.8.13.3 Generating and Propagating Web Server Plug-In

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

1. In the tree to the left, expand **Servers** and select **Web servers.**
2. Check the **RTDWebServer** checkbox.
3. Click **Generate Plug-in**.
4. Click **Propagate Plug-in**.

13.8.13.4 Starting the Web Server

To start the web server, perform the following steps:

1. In the tree to the left, expand **Servers** and select **Web servers**.
2. Check the **RTDWebServer** checkbox.
3. Click **Start**.

   The log directory is `C:\Program Files\IBM\HTTPServer\logs\`.

   Look in `access.log` and `error.log`.

   This creates an IBM HTTP Web Server with round-robin load balancing on port 8080.

   See C:\Program Files\IBM\HTTPServer\Plugins\config\RTDWebServer\`plugin-cfg.xml`.

13.8.14 Starting the Cluster and Servers

To start the cluster and servers, perform the following steps:

1. In the tree to the left, expand **Servers** and select **Clusters**.
2. Check the cluster checkbox **RTDCluster**.
3. Click **Start**.
4. Wait until the cluster has started.
5. In the tree to the left, expand **Servers** and select **Application servers**.
6. Verify all six servers have started: look for green arrows in the **Status** column.

   The base log directory for Machine #1 and Machine #2 is `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\logs\`.

   - Look in `\RTDServer1a\SystemOut.log` for **Server RTDServer1a open for e-business**.
   - Look in `\RTDServer1b\SystemOut.log` for **Server RTDServer1b open for e-business**.
   - Look in `\RTDServer1c\SystemOut.log` for **Server RTDServer1c open for e-business**.
   - Look in `\RTDServer2a\SystemOut.log` for **Server RTDServer2a open for e-business**.

- Look in `\RTDServer2b\SystemOut.log` for **Server RTDServer2b open for e-business**.
- Look in `\RTDServer2c\SystemOut.log` for **Server RTDServer2c open for e-business**.

13.8.15 Deploying RTD.ear onto Cluster

This process consists of deploying `RTD.ear` to the cluster, generating and propagating the web server plug-in, then starting the OracleRTD application.

13.8.15.1 Deploying RTD.ear to Cluster

To deploy `RTD.ear` to the cluster, perform the following steps:

1. In the tree to the left, expand **Applications** and select **Install New Application**.
2. For **Full path**, click **Browse...** and browse to `*RTD_HOME*/package/RTD.ear`.

   For example, `C:/OracleBI/RTD/package/RTD.ear`.
3. Click **Next**.
4. Step 1: For **Application name**, enter `OracleRTD.`
5. Step 1: Click **Next**.
6. Step 2: In **Clusters and Servers:**, select **cluster=RTDCluster**.
7. Step 2: Click the **Select All** icon.
8. Step 2: Click **Apply**.
9. Step 2: Click **Next**.
10. Step 3: Click **Finish**.
11. Click **Save**.

13.8.15.2 Generating and Propagating Web Server Plug-In

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

1. In the tree to the left, expand **Servers** and select **Web servers**.
2. Check the **RTDWebServer** checkbox.
3. Click **Generate Plug-in**.
4. In the tree to the left, expand **Servers** and select **Web servers**.
5. Check the **RTDWebServer** checkbox.
6. Click **Propagate Plug-in**.

13.8.15.3 Starting the OracleRTD Application

To start the OracleRTD application, perform the following steps:

1. In the tree to the left, expand **Applications** and select **Enterprise Applications**.
2. Check the **OracleRTD** checkbox.
3. Click **Start**.
4. Wait for the startup to complete.

The base log directory for Machine #1 and Machine #2 is `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\logs\`. Specific log files are as follows:

- `\RTDServer1a\SystemOut.log`
- `\RTDServer1b\SystemOut.log`
- `\RTDServer1c\SystemOut.log`
- `\RTDServer2a\SystemOut.log`
- `\RTDServer2b\SystemOut.log`
- `\RTDServer2c\SystemOut.log`

For both Machine #1 and Machine #2, other log information appears under the directory `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01\installedApps\`*`host_nameNode_name`*`\`. Look in `\OracleRTD\logs\server.log`.

13.8.16 Setting Classloader Priority

To give priority to the application classloader over the application server classloader, after you have deployed Oracle RTD, you must explicitly set the Classloader priority in the Websphere administrative console, as follows:

1. Log into the WebSphere administrative console, using the administrative user and password.
2. Expand **Applications**, then choose **Enterprise Applications**.
3. Click **OracleRTD**.
4. Click **Class loading and update detection**.
5. Select **Classes loaded with application class loader first**.
6. Click OK.
7. Restart WebSphere.

13.8.17 Viewing and Changing User-Role Associations

The standard Oracle RTD Roles and Roles-groups mapping have been predefined for WebSphere.

To show the Oracle RTD Roles and Roles-groups mapping, perform the following steps:

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**.

If you want to view or edit the user-role associations, check the role to be modified, then click either **Look up users** or **Look up groups** and change the mapping.

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

This section consists of the following topics:

- Section 13.8.18.1, "Creating Custom Roles"

- Section 13.8.18.2, "Assigning Permissions to Custom Roles"

13.8.18.1 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 13.8.10.2, "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 13.8.18.1.1, "Specifying Roles in application.xml" for details.

3. Map the roles to the WebSphere groups in the Integrated Solutions Console. See Section 13.8.18.1.2, "Mapping Roles to Groups" for details.

Then, perform either of the following:

1. Uninstall, then redeploy Oracle RTD, as follows:

   a. Download the two deployment descriptor files, `application.xml` and `ibm-application-bnd.xmi`, back into `RTD.ear` (the file `ibm-application-bnd.xmi` contains the role-to-group mappings).

   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 13.8.18.1.1, "Specifying Roles in application.xml"
- Section 13.8.18.1.2, "Mapping Roles to Groups"

13.8.18.1.1 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 id="SecurityRole_1241469153092">
           <role-name>ILS2Users</role-name>
       </security-role>
   ```

   where `security-role id` is any unique value.

2. Repeat step 1 for as many roles as you want to create.

13.8.18.1.2 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.

> **Note:** To view and change the user-role associations, see Section 13.8.17, "Viewing and Changing User-Role Associations."

13.8.18.2 Assigning Permissions to Custom Roles

As described in Section 7.4, "Assigning Permissions," assign Cluster permissions, Inline Service permissions, and Decision Center Perspective permissions to any custom roles.

13.8.19 Creating startJConsole Batch Files

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

13.8.19.1 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 **RTDServer1a**.
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.
5. Repeat steps 1-4 for servers **RTDServer1b**, **RTDServer1c**, **RTDServer2a**, **RTDServer2b**, and **RTDServer2c.**

13.8.19.2 Creating JConsole Startup Script For Each Server

To create the JConsole startup script for each server, perform the following steps:

1. On machine #1, create and open a new file in Notepad for the first server instance RTDServer1a.
2. Add the following lines into the file:

   For Windows:

   ```
   call c:\program
   files\ibm\websphere\appserver\profiles\appsrv01\bin\setupCmdL
   ine
   ```

   ```
   "%JAVA_HOME%\bin\jconsole" "-J-Djava.class.path=%WAS_
   HOME%\runtimes\com.ibm.ws.admin.client_6.1.0.jar;%JAVA_
   HOME%\lib\tools.jar" "-J%CLIENTSSL%" "-J%CLIENTSAS%"
   service:jmx:iiop://HOST:PORT/jndi/JMXConnector
   ```

   For Linux:

   ```
   #!/bin/sh

   WAS_HOME=/usr/IBM/WebSphere/AppServer

   USER_HOME=/usr/IBM/WebSphere/AppServer/profiles/AppSrv01

   WAS_HOST=MACHINE_HOST

   WAS_BOOTSTRAP_PORT=BOOTSTRAP_ADDRESS
   ```

```
$WAS_HOME/java/bin/jconsole -J-Djava.class.path=$WAS_
HOME/runtimes/com.ibm.ws.admin.client_6.1.0.jar:$WAS_
HOME/java/lib/tools.jar
-J-Dcom.ibm.CORBA.ConfigURL=file:$USER_
HOME/properties/sas.client.props
-J-Dcom.ibm.SSL.ConfigURL=file:$USER_
HOME/properties/ssl.client.props service:jmx:iiop://$WAS_
HOST:$WAS_BOOTSTRAP_PORT/jndi/JMXConnector
```

3. For the Linux script, replace *BOOTSTRAP_ADDRESS* with the port number seen in step 4 of Section 13.8.19.1, "Getting Bootstrap Addresses For Each Server." For example, `2810`.
4. For the Linux script, replace *MACHINE_HOST* with the machine host name for server RTDServer1a, for example, `mymachine.us.oracle.com`.
5. Save the file as `RTDServer1a_JConsole.bat` for Windows or `RTDServer1a_JConsole.sh` for Linux.
6. Repeat steps 1-5 for servers **RTDServer1b**, **RTDServer1c**, **RTDServer2a**, **RTDServer2b**, **RTDServer2c**, using their corresponding ports and machine host names.

13.8.20 Deploying CrossSell Inline Service

To deploy the CrossSell Inline Service, perform the following steps:

1. On machine #1, start Decision Studio by running *RTD_HOME*`\eclipse\eclipse.exe`.
2. In top menu bar, click **File** then **Import**.
3. Select **Existing Projects into Workspace**.
4. Click **Next**.
5. Click **Browse...**
6. Go to *RTD_HOME*`\examples\CrossSell`.
7. Click **OK**.
8. Click **Finish**.
9. Wait until the Inline Service build is finished.
10. In top menu bar, click the icon **Deploy the Inline Service to a server**.
11. Click **Deploy**.

Note: By default, Decision Studio will deploy to **localhost:8080**, which is machine #1 and the web server "load balancer" port.

12. For **User Name**, enter `sdsu`.
13. Click **Connect**.
14. Wait until the CrossSell Inline Service is deployed.

13.8.21 Creating CrossSell Database Tables

To create the CrossSell database tables, perform the following steps:

1. On machine #1, in a command console:
   cd to `RTD_HOME\examples\CrossSell\etc\data\Oracle`
2. Run the following command:

   ```
   InitAppDB.cmd RTD_HOME db_host_name port sid db_user_name
   admin_user_name admin_user_password
   ```

   For example:

   ```
   InitAppDB.cmd C:\OracleBI\RTD mydbmachine.us.oracle.com 1521
   orcl MYUSERNAME MYADMINUSERNAME MYADMINPASSWORD
   ```

   > **Note:** If the database is on a SQL Server named instance, specify the *db_host_name* parameter using the format *host_name\instance_name*.

13.8.22 Running Load Generator Against CrossSell Inline Service

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

1. On machine #1, start Oracle RTD Load Generator by running:

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

   Select **clientHttpEndPoints.properties** (which is already set to localhost:8080 or machine #1).
8. Click **Open**.
9. In the top menu bar, click **File,** then **Save**.
10. In the top menu bar, click the icon **Runs the current Load Generator script** (blue right arrow).

13.8.23 Creating WebSphere Thread Dump

To create a WebSphere thread dump, perform the following steps:

1. On machine #1, open a command window.
2. Go to `C:\Program Files\IBM\WebSphere\AppServer\bin` and run `wsadmin.bat.`
3. Enter `set jvm [$AdminControl completeObjectName type=JVM,process=RTDServer1a,*]`, replacing RTDServer1a for each server to analyze.
4. Enter `$AdminControl invoke $jvm dumpThreads`.

5. Go to `C:\Program Files\IBM\WebSphere\AppServer\profiles\AppSrv01` and view the thread dump file created.

13.9 Setting up an Oracle AS Cluster

This section consists of the following topics:

- Section 13.9.1, "Cluster Machine Setup"
- Section 13.9.2, "Operations to Perform on Host1 and Host2"
- Section 13.9.14, "Joining the Clusters on Host1 and Host2"

13.9.1 Cluster Machine Setup

This section describes how to install Oracle RTD onto two hosts, each running a cluster of OC4J instances.

In general, three types of clusters are supported, corresponding to different distributions of Oracle RTD services:

- DS (Decision Service)
- DC (running Decision Center's UI servlet and Studio's soap servlet)
- LS (Learning Service)

The Oracle RTD cluster types have the following properties:

- Type 1 cluster: a homogeneous cluster of at least two Oracle RTD instances, all running DS, DC, and LS.

  Either all the instances are on one physical host or the instances are split across two physical hosts.

- Type 2 cluster: a cluster of at least 4 Oracle RTD instances, two running DS and DC, and the others running LS.

  Either all the instances are on one physical host or the two types of instance are split across two physical hosts.

- Type 3 cluster: a cluster of at least 4 Oracle RTD instances, two running just DS, and the others running DC and LS.

  Either all the instances are on one physical host or the two types of instance are split across two physical hosts.

This section concentrates on types 2 and 3, as these have the most detailed setup procedures.

For both cluster type 2 and cluster type 2, configuration starts on Host1, with 4 OC4J instances deployed into two OC4J groups, as follows:

- rtdGroup

  Instances:

  – rtd_1_1 on Host1

  – rtd_1_2 on Host1

- rtdGroup2

  Instances:

  - rtd_1_3 on Host1
  - rtd_1_4 on Host1

[Note the optional name convention: rtd_<*host_number*>_<*instance_within_host*>.]

Assuming that you want to deploy on multiple hosts, you then configure a cluster on Host2 using the same group names as on Host, as follows:

- rtdGroup

  Instances:

  - rtd_2_1 on Host2
  - rtd_2_2 on Host2
- rtdGroup2

  Instances:

  - rtd_2_3 on Host2
  - rtd_2_4 on Host2

Then you merge the two clusters into one cluster managed by Host1:

- rtdGroup

  Instances:

  - rtd_1_1 on Host1
  - rtd_1_2 on Host1
  - rtd_2_1 on Host2
  - rtd_2_2 on Host2
- rtdGroup2

  Instances:

  - rtd_1_3 on Host1
  - rtd_1_4 on Host1
  - rtd_2_3 on Host2
  - rtd_2_4 on Host2

13.9.2 Operations to Perform on Host1 and Host2

The following sections describe the operations which are to performed on both Host1 and Host2:

- Section 13.9.3, "Installing Oracle AS 10.1.3.3"
- Section 13.9.4, "Optionally Changing Port"
- Section 13.9.5, "Creating OC4J Groups and Instances"
- Section 13.9.6, "Creating JDBC Resources"
- Section 13.9.7, "Setting Server Properties"
- Section 13.9.8, "Creating Oracle AS RTD Data Source"
- Section 13.9.9, "Editing OC4J Instance Configurations"
- Section 13.9.10, "Deploying RTD.ear into rtdGroup as OracleRTD"

- Section 13.9.11, "Deploying RTD.ear into rtdGroup2 as OracleRTD2"
- Section 13.9.12, "Verifying the Configuration"
- Section 13.9.13, "Disabling Certain Oracle RTD Services"

13.9.3 Installing Oracle AS 10.1.3.3

After you have completed the installation process, Oracle AS will be started.

You will perform many of the Oracle AS administration tasks in the Enterprise Manager, `ascontrol`. The launch URL for the Enterprise Manager is installation dependent, but is typically `http://localhost:7777/em`.

13.9.4 Optionally Changing Port

After stopping Oracle AS, you can change the port by editing the file *`OAS_HOME`*`/Apache/Apache/conf/httpd.conf`. A typical location for *`OAS_HOME`* is `C:/product/10.1.3.1/OracleAS_1/`.

1. To stop Oracle AS, run *`OAS_HOME`*`/opmn/bin/opmnctl.exe stopall`.
2. Locate and open the file `httpd.conf` (for example, C:/product/10.1.3.1/OracleAS_1/`Apache/Apache/conf/httpd.conf`).
3. Search for the word **Port** in the file.
4. Replace the **Port** and **Listen** values by `8080`.

13.9.5 Creating OC4J Groups and Instances

To create OC4J groups and instances, perform the following steps:

1. In the Enterprise Manger, create two OC4J groups: **rtdGroup** and **rtdGroup2**.
2. Create four OC4J instances on each host, as follows:
   - For Host 1, create instances **rtd_1_1** and **rtd_1_2** in rtdGroup
   - For Host 2, create instances **rtd_2_1** and **rtd_2_2** in rtdGroup
   - For Host 1, create instances **rtd_1_3** and **rtd_1_4** in rtdGroup2
   - For Host 2, create instances **rtd_2_3** and **rtd_2_4** in rtdGroup2

   [Note the instance name convention used: rtd_<*host_number*>_<*instance_within_host*>.]

13.9.6 Creating JDBC Resources

You create JDBC resources after creating the OC4J groups because when you create JDBC connection pools and data sources at the group level, they are automatically copied to each of the group's OC4J instances that exists at the time.

Later, however, when you add an OC4J instance to a group, configuration for the data sources you had created at the group level are not automatically distributed to the new OC4J instance.

To create JDBC resources, perform the following steps:

1. If you will be using SQLServer or DB2, first copy the corresponding jdbc jars provided with `rtd_3.0.0_OC4J_win.zip` (or `rtd_3.0.0_OC4J_unix.cpio` if on Linux/Unix) into each OC4J instance's applib directory, as follows:

a. Stop Oracle AS.

b. For each OC4J instance (rtd_1_1, rtd_1_2, and so on), copy the appropriate file or files:

   (For SQLServer) `RTD_HOME/lib/jdbc/sqljdbc.jar`

   (For DB2) `RTD_HOME/lib/jdbc/db2jcc.jar`

   `RTD_HOME/lib/jdbc/db2jcc_license_cu.jar`

   into `OAS_HOME/j2ee/OC4J_INSTANCE/applib/`.

c. Restart Oracle AS.

2. In the Enterprise Manager, drill into **rtdGroup**.

3. Go to the **Administration** tab, then click the **JDBC Resources** task under the Services heading. Under the Connection Pools heading, click **Create** to create a new connection pool for your Oracle RTD Database.

   If you are using OC4J as part of Oracle Application Server, first click **home** under the Groups heading, then proceed to the **Administration** tab.

4. On the Create Connection Pool - Application page, ensure that **default** is selected for **Application**, and that **New Connection Pool** is selected for **Connection Pool Type**. Then, click **Continue**.

5. On the Create Connection Pool page, enter `RTDConnectionPool` for **Name**.

6. For Connection Factory Class, enter one of the following:

   - **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerDriver`
   - **Oracle Database:** `oracle.jdbc.pool.OracleDataSource`
   - **DB2:** `com.ibm.db2.jcc.DB2Driver`

7. For JDBC URL, enter one of the following:

   - **SQL Server:** `jdbc:sqlserver://`*`db_host`*`:`*`db_port`*`;databaseName=`*`db_name`*

   > **Note:** If the database is on a SQL Server named instance, specify the *`db_host`* parameter using the format *`host_name\instance_name`*.

   - **Oracle Database:** `jdbc:oracle:thin:@`*`db_host`*`:`*`db_port`*`:`*`sid`*
   - **DB2:** `jdbc:db2://`*`db_host`*`:`*`db_port`*`/`*`db_name`*

   `db_host` is the name of the server running the Oracle RTD Database, *`db_port`* is the port number for connecting to the database server, *`db_name`* is the name of the Oracle RTD Database instance (such as `rtd`), and *`sid`* is the Oracle System Identifier that refers to the instance of the Oracle Database running on the server.

8. Under the Credentials heading, for **Username**, provide a database user name with system administration privileges for the Oracle RTD Database instance. Then, provide the corresponding password. See *Oracle Fusion Middleware Security Guide* for information about whether to provide a cleartext password or an indirect password.

9. Click **Finish**.

10. On the JDBC Resources page, under the Data Sources heading, click **Create** to define a new data source.

11. On the Create Data Source - Application & Type page, ensure that **default** is selected for **Application**, and that **Managed Data Source** is selected for **Data Source Type**. Then, click **Continue**.
12. On the Create Data Source - Managed Data Source page, enter `RTD_DS` for **Name**, `jdbc/SDDS` for **JNDI Location**, and select **RTDConnectionPool** for **Connection Pool**. Keep the defaults for the other options. Then, click **Finish**.
13. On the JDBC Resources page, in the Data Sources table, click **Test Connection** for the **RTD_DS** data source. Follow these steps to test the RTD_DS data source:
    - If you are using Oracle Database for your Oracle RTD Database, keep the default settings and click **Test**.
    - If you are using SQL Server or DB2 for your Oracle RTD Database, change the SQL Statement to `select * from SDAPPS`, then click **Test**.

    If the connection is not established successfully, restart OC4J and then test the data source again. If it still fails, ensure that your connection pool settings are correct.

The JDBC configuration that you have just created for rtdGroup will be copied to rtdGroup2, after you stop Oracle AS, as follows:

- In *`OAS_HOME`*`/opmn/bin`, run `opmnctl.exe stopall`

13.9.7 Setting Server Properties

To set server properties, edit the file *`OAS_HOME`*`/opmn/conf/opmn.xml`.

For each rtd instance, perform the following tasks:

- Adjusting Start Parameters
- Changing Web-Site
- Checking Process Group

13.9.7.1 Adjusting Start Parameters

To adjust start parameters, perform the following steps:

1. Add the following to the start parameters, all on one line:

    -
      ```
      -Drtd.instanceName=rtd_1_1
      -DSDGroupName=rtdDon
      -DRestrictClusterMembers=false
      -Dcom.sun.management.jmxremote.authenticate=false
      -Dcom.sun.management.jmxremote.ssl=false
      -Dcom.sun.management.jmxremote=true
      -Dcom.sun.management.jmxremote.port=12351
      -Djava.net.preferIPv4Stack=true
      ```

    For `rtd.instanceName`, use the appropriate OC4J instance name (rtd_1_1, rtd_1_2, and so on). If you are not using the suggested instance name convention (rtd_*<host_number>*_*<instance_within_host>*), use any alphanumeric string that is a valid file name, with no space or special characters.

    For `SDGroupName`, use the same name for all OC4J instances in the cluster, across all hosts. Ensure that the name uses alphanumeric characters without spaces or punctuation.

    For `jmxremote.port`, use a distinct port for each instance in this host.

> **Tip:** Let the last digit of the port (as in the examples 1235**1**, 1235**2**, 1235**3**, and so on) represent the instance number within the port.

Note that the jmx properties are optional. If you prefer using Enterprise Manager to access the OracleRTD MBeans instead of JConsole, do not add the following properties:

-
  ```
  -Dcom.sun.management.jmxremote.authenticate=false
  -Dcom.sun.management.jmxremote.ssl=false
  -Dcom.sun.management.jmxremote=true
  -Dcom.sun.management.jmxremote.port=12351
  ```

2. Reduce memory.

   You may want to let each OC4J instance start with less memory, including j2ee/home, the one that runs `ascontrol`.

   Add or change the following options to the java start options, in the same string as the system properties listed in step 1 of this section:

   - `-Xmx1024M`

     `-Xms128M`

     `-XX:MaxPermSize=128M`

3. Disable Learning Service or Decision Service.

   > **Note:** This step is optional at this point, because you can perform the same task through JConsole after starting the Oracle RTD instances, as described in Section 13.9.13, "Disabling Certain Oracle RTD Services."

   If you choose to disable Learning Service or Decision Service through system properties, add the following properties to `opmn.xml`:

   -
     ```
     -DLearningServiceEnabled=false
     -DDecisionServiceEnabled=false
     ```

13.9.7.2 Changing Web-Site

Change the referenced web-site from `default-web-site` to `rtd-web-site`. For example:

- `<port id="`**`rtd-web-site`**`" range="12501-12600" protocol="ajp"/>`

13.9.7.3 Checking Process Group

Ensure that `opmn.xml` has the right group specification for each OC4J instance.

The last element inside `<process-type>` is the `process-set` element. Check that the `process-set id` value is the same as the OC4J group name, as in the following example:

- `<process-set id="`**`rtdGroup`**`" numprocs="1"/>`

13.9.8 Creating Oracle AS RTD Data Source

For each of the OC4J instances in rtdGroup, you should see a file:

`OAS_HOME``/j2ee/`*`OC4J_INSTANCE`*`/config/data-sources.xml`

These should have been created in the previous procedures, and contain something similar to the following:

```
<?xml version = '1.0' encoding = 'UTF-8'?>

<data-sources
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracleas/
schema/data-sources-10_1.xsd" schema-major-version="10"
schema-minor-version="1">

<managed-data-source connection-pool-name="RTDConnectionPool"
jndi-name="jdbc/SDDS" name="RTD_DS"/>

<connection-pool name="RTDConnectionPool">

<connection-factory
factory-class="com.microsoft.sqlserver.jdbc.SQLServerDriver"
user="userName" password="myPassword"
url="jdbc:sqlserver://myhost:1433;databaseName=mydb"/>

</connection-pool>

</data-sources>
```

Copy one of these to the OC4J instances in the other OC4J group, rtdGroup2.

13.9.9 Editing OC4J Instance Configurations

For each OC4J instance, perform the following tasks:

- Editing server.xml
- Creating rtd-web-site.xml

13.9.9.1 Editing server.xml

In the file *OAS_HOME*/j2ee/*OC4J_INSTANCE*/config/server.xml, change:

- `<web-site default="true" path="./default-web-site.xml" />`

to

- <web-site default="true" path="./**rtd-web-site.xml**" />

13.9.9.2 Creating rtd-web-site.xml

To create rtd-web-ste.xml, perform the following steps:

1. Create the file *OAS_HOME*/j2ee/*OC4J_INSTANCE*/config/rtd-web-site.xml, by copying default-web-site.xml in the same directory.
2. Delete all of its <web-app> elements (others will be added automatically when you deploy OracleRTD).
3. Leave the port set to 0, since this will be adjusted automatically when OPMN starts the OC4J instance.
4. Leave the protocol set to ajp. This is required for clustering.

13.9.10 Deploying RTD.ear into rtdGroup as OracleRTD

To deploy RTD.ear into rtdGroup as OracleRTD, perform the following steps:

1. Start Oracle AS, by running `OAS_HOME/opmn/bin/opmnctl.exe startall`.
2. Login to Enterprise Manager: `http://localhost:7777/em`.
3. Drill into the OC4J group **rtdGroup**.
4. Select its Application tab.
5. Click **Deploy**.
6. Select **RTD.ear** in the location where you unzipped it.

   The deployment plan strategy to adopt is as follows:

   - For Oracle RTD Type 1 and Type 2 clusters, let the deployment create its own deployment plan
   - For Oracle RTD Type 3 clusters, let the first-time deployment create its own deployment plan, save the plan, then use it on subsequent deployments
7. Click **Next** to upload RTD.ear.
8. Enter `OracleRTD` as the application name.

   This is required because the file `orion-application.xml` provided with Oracle RTD refers to its containing directory as OracleRTD.
9. Select **rtd-web-site** as the target web-site.

   Note: This is the main point at which the deployment varies according to cluster type, where you need to specify which Oracle RTD services need to be dispatchable through the public OHS (Oracle HTTP Server) port.

   The requirement is for OHS (Oracle HTTP Server) to dispatch requests from Decision Center (/ui), Studio (/soap), and Decision Service clients (/rtis) to the group where these services are activated. To achieve this, the next step sets up alternative web context roots for the deactivated services.
10. For type 1 and 2 clusters, leave the web context-roots as they are, namely:

    - /ui, /ls, /soap, /schema, /rtis

    For type 3 (DS only) clusters, during the first-time deployment, adjust the proposed web context-roots by appending "-ds" to each of them except /rtis, as follows:

    - /ui-ds, /ls-ds, /soap-ds, /schema-ds, /rtis

    Save the deployment plan for subsequent deployments.

    Tip: Give the plan a name that indicates its function, such as `dsPlan.dat` for a DS-only cluster.
11. Click **Next** to accept the deployment plan.
12. Click **Next** to start the deployment.

13.9.11 Deploying RTD.ear into rtdGroup2 as OracleRTD2

The steps here are the same as for rtdGroup, except for the following:

- The web-app bindings will be different.
- The application name must different (OracleRTD2 versus OracleRTD) because OHS keys its routing instructions based on application name.

1. In the Enterprise Manager, drill into OC4J group **rtdGroup2**.
2. Select its Application tab.
3. Click **Deploy**.
4. Select **RTD.ear** in the location where you unzipped it.
5. Click **Next** to upload RTD.ear.
6. Enter `OracleRTD2` as the application name.
7. Select **rtd-web-site** as the target web-site.
8. Click **Next**.
9. Edit the web context-roots as follows (for first-time deployment only):

   For Type 2 (LS only) clusters:

   a. Adjust the proposed web context-roots by appending "-2" to each of them:

      /ui-2, /ls-2, /soap-2, /schema-2, /rtis-2

   b. Edit the deployment plan:

      Change the classpath settings to refer to the new application name `${oracle.j2ee.home}/OracleRTD2` instead of `${oracle.j2ee.home}/OracleRTD`.

   c. Save the deployment plan for subsequent deployments, for example, naming it `lsPlan.dat`.

   For Type 3 (DC and LS) clusters:

   a. Adjust the proposed web context-roots by appending "-2" to /rtis:

      /ui, /ls, /soap, /schema, /rtis-2

   b. Edit the deployment plan:

      Change the classpath settings to refer to the new application name `${oracle.j2ee.home}/OracleRTD2` instead of `${oracle.j2ee.home}/OracleRTD`.

   c. Save the deployment plan for subsequent deployments, for example, naming it `dcLsPlan.dat`.

10. Click **Next** to start the deployment.

13.9.12 Verifying the Configuration

To verify the configuration, perform the following steps:

1. Stop Oracle AS.
2. Verify rtd-web-site.xml.

   rtdGroup Verification

   In the rtd_1_1 and rtd_1_2 instances, there should be web-app entries bound to /ui, /schema, and so on:

   ```
   <?xml version="1.0"?>
   ```

```
<web-site
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracle
as/schema/web-site-10_0.xsd" port="12552" protocol="ajp13"
display-name="OC4J 10g (10.1.3) RTD Web Site"
schema-major-version="10" schema-minor-version="0" >

<default-web-app application="default" name="defaultWebApp"
root="/j2ee" />

<web-app application="OracleRTD" name="ui"
load-on-startup="true" root="/ui" />

<web-app application="OracleRTD" name="schema"
load-on-startup="true" root="/schema" />

<web-app application="OracleRTD" name="soap"
load-on-startup="true" root="/soap" />

<web-app application="OracleRTD" name="ls"
load-on-startup="true" root="/ls" />

 <web-app application="OracleRTD" name="rtis"
load-on-startup="true" root="/rtis" />

<access-log path="../log/default-web-access.log" split="day"
/>

 </web-site>
```

rtdGroup2 Verification

In the rtd_1_3 and rtd_1_4 instances, there should be web-app entries bound to /ui-ls, /schema-ls, and so on:

```
<?xml version="1.0"?>

<web-site
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://xmlns.oracle.com/oracle
as/schema/web-site-10_0.xsd" port="12553" protocol="ajp13"
display-name="OC4J 10g (10.1.3) RTD Web Site"
schema-major-version="10" schema-minor-version="0" >

<default-web-app application="default" name="defaultWebApp"
root="/j2ee" />

<web-app application="OracleRTD" name="ui"
load-on-startup="true" root="/ui-ls" />

<web-app application="OracleRTD" name="schema"
load-on-startup="true" root="/schema-ls" />

<web-app application="OracleRTD" name="soap"
load-on-startup="true" root="/soap-ls" />

<web-app application="OracleRTD" name="ls"
load-on-startup="true" root="/ls-ls" />

<web-app application="OracleRTD" name="rtis"
load-on-startup="true" root="/rtis-ls" />

<access-log path="../log/default-web-access.log" split="day"
/>
```

```
</web-site>
```

The port assignment is not important, as it happens automatically.

13.9.13 Disabling Certain Oracle RTD Services

Use JConsole or other MBean browsers to disable Decision Service or Learning Service in certain Oracle RTD instances according to the cluster type. This only needs to be done when changing cluster types. It is not necessary if redeploying Oracle RTD into the same cluster type.

> **Note:** The instructions for type 2 and type 3 clusters are currently the same. So, if you switch from a type 2 to a type 3 cluster, or vice versa, no actions are required in this section.

1. Start JConsole and connect to each instance. For each instance, do the following:

   Cluster Type 1: No changes are required

   Cluster Type 2

   a. rtdGroup (runs DS and DC)

      For instances rtd_1_1, rtd_1_2, rtd_2_1, rtd_2_2, disable LS:

      - Navigate: OracleRTD > SDPropertyManager > Misc

      - Set: `LearningServiceEnabled = false`

   b. rtdGroup2 (runs LS)

      For instances rtd_1_3, rtd_1_4, rtd_2_3, rtd_2_4, disable DS:

      - Navigate: OracleRTD > SDPropertyManager > Misc

      - Set: `DecisionServiceEnabled = false`

   Cluster Type 3

   a. rtdGroup (runs DS)

      For instances rtd_1_1, rtd_1_2, rtd_2_1, rtd_2_2, disable LS:

      - Navigate: OracleRTD > SDPropertyManager > Misc

      - Set: `LearningServiceEnabled = false`

   b. rtdGroup2 (runs DC and LS)

      For instances rtd_1_3, rtd_1_4, rtd_2_3, rtd_2_4, disable DS:

      - Navigate: OracleRTD > SDPropertyManager > Misc

      - Set: `DecisionServiceEnabled = false`

2. Set Decision Service Address.

   For Type 3 clusters, where DC and DS are not co-located, use JConsole to set the cluster's Decision Service address, so that DC can send test events from its Interactive Integration Map view.

   - Navigate: OracleRTD > SDPropertyManager > Misc

   - Set: `DecisionServiceAddress = http://myHost.com:8080`

13.9.14 Joining the Clusters on Host1 and Host2

In this example, start first with host2 first, then host1, so that host1 will be running `ascontrol` at the end of the process.

13.9.14.1 Combining Two Separate Oracle Application Server Installations Into a Cluster

Suppose you installed two separate Oracle Application Server instances (as1 on Host1 and as2 on Host2) and during the installation of both instances, you selected the option to include an Administration OC4J. You did not, however, select the option to define a multicast address and join an existing Oracle Application Server cluster.

In this scenario, each Oracle Application Server instance is running a separate Application Server Control.

To combine these two instances into one cluster topology so they are managed by one Cluster Topology page, which is displayed by one active Application Server Control, perform the following steps.

1. Open your Web browser and enter the Application Server Control URL for as1.
2. Log in to the Application Server Control, and scroll to the Administration section of the Cluster Topology page.
3. Click **Topology Network Configuration** to display the Topology Network Configuration page.

Note: On the Topology Network Configuration page:

- The current application server instance is selected in the View By field.
- The fields in the Topology section of the page are empty. This indicates that the application server instance does not belong to a cluster.

4. Select **Configuring Dynamic Node Discovery Using Multicast** and enter a multicast address and port number in the Discover field.

   For example: `229.1.2.50:5555`

   The multicast address must be within the following range: 224.0.0.1 to 239.255.255.255. The port can be any four-digit number. In the configuration file, the multicast address must be preceded by an asterisk (*), but when you enter the address in this field, Application Server Control automatically includes the asterisk if you do not specify it here.
5. Make a note of the multicast address and port, and then click **Apply**.
6. Enter the Application Server Control URL for as2.
7. Repeat steps 2 through 5 for as2.

   Be sure to use the same multicast address and port that you used when you configured as1.
8. Navigate to the Cluster Topology page.

   Note that both Oracle Application Server instances now appear on the Cluster Topology page.

9. Click **Expand All** and then select the ascontrol application (which represents the Application Server Control) that is deployed to the as1 instance.
10. Click **Stop** to stop the selected ascontrol application.

    Note: Do not stop the active ascontrol application, but stop any other ascontrol applications that are up and running. There is no need to run two Application Server Control applications in the same cluster.

13.10 Setting up a JBoss Cluster

This section describes the JBoss clustering install instructions, assuming that you have two machines to be set up as follows:

- Machine #1
  - Oracle RTD software
  - JBoss Application Server
  - Apache HTTP Server
- Machine #2
  - JBoss Application Server
  - Oracle RTD JDBC JAR and SSL Files

The rest of this section consists of the following topics:

- Section 13.10.1, "Installing the Oracle RTD Software"
- Section 13.10.2, "Initializing the Oracle RTD Database"
- Section 13.10.3, "Setting Up and Installing the Apache Software"
- Section 13.10.4, "Installing the JBoss Software"
- Section 13.10.5, "Configuring the JBoss Software"
- Section 13.10.6, "Creating Oracle RTD Roles and Users"
- Section 13.10.7, "Starting the JBoss Application Server"
- Section 13.10.8, "Deploying RTD.ear to the JBoss Cluster"
- Section 13.10.9, "Undeploying RTD.ear from the JBoss Cluster"
- Section 13.10.10, "Stopping the JBoss Application Server"
- Section 13.10.11, "JBoss and Oracle RTD Logs"

13.10.1 Installing the Oracle RTD Software

To install the Oracle RTD software, perform the following steps:

1. On Machine #1, install Oracle RTD from `rtd_3.0.0_jboss_win.zip` (or `rtd_3.0.0_jboss_unix.cpio` if on Linux/Unix) to *`RTD_HOME`*, for example, `C:\OracleBI\RTD`.
2. On Machine #2, create the following directory: *`RTD_HOME`*`\lib\jdbc\`.
3. On Machine #1, copy the jars in *`RTD_HOME`*`\lib\jdbc\` to *`RTD_HOME`*`\lib\jdbc\` on Machine #2.
4. On Machine #2, create the following directory: *`RTD_HOME`*`\etc\ssl\`.

5. On Machine #1, copy the file `RTD_HOME\etc\ssl\sdserver.keystore` to `RTD_HOME\etc\ssl\` on Machine #2.
6. On Machine #2, create the following directory: `RTD_HOME\log\`.

13.10.2 Initializing the Oracle RTD Database

To initialize the Oracle RTD database on Machine #1, run the following script:

- `RTD_HOME \scripts\SDDBTool.cmd`

  For more information, see Section 2.2.4, "Initializing the Oracle RTD Database Using SDDBTool."

13.10.3 Setting Up and Installing the Apache Software

This section consists of the following topics:

- Section 13.10.3.1, "Installing the Apache Server and Downloading mod_jk"
- Section 13.10.3.2, "Configuring the Apache Server"
- Section 13.10.3.3, "Creating the File mod_jk.conf"
- Section 13.10.3.4, "Creating the File uriworkermap.properties"
- Section 13.10.3.5, "Creating the File workers.properties"
- Section 13.10.3.6, "Starting the Apache HTTP Server"

13.10.3.1 Installing the Apache Server and Downloading mod_jk

To install the Apache server and to download mod_jk, perform the following steps:

1. On Machine #1, download Apache HTTP Server 2.2.6 (`apache_2.2.6-win32-x86-no_ssl.msi`) from `http://archive.apache.org/dist/httpd/binaries/win32/`.
2. On Machine #1 in the cluster, install Apache HTTP Server 2.2.6 to *`APACHE_INSTALL_DIR`*, for example, `C:\Program Files\`.

   Terminology: From this point on, *`APACHE_HOME_DIR`* refers to *`APACHE_INSTALL_DIR`*`\Apache Software Foundation\Apache2.2`

   ---

3. Download Mod JK 1.2 (`mod_jk-1.2.27-httpd-2.2.10.so`) from `http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/win32/jk-1.2.27/`.
4. Rename it to `mod_jk.so` and copy it to *`APACHE_HOME_DIR`*`\modules\`.

13.10.3.2 Configuring the Apache Server

To configure the Apache server, perform the following steps:

1. On Machine #1, open *`APACHE_HOME_DIR`*`\conf\httpd.conf`.

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

2. Change the port for the property **Listen** to `8080`.

   For example: `Listen 8080`

3. Change the port for the property **ServerName** to `8080`.

   For example: `ServerName myHostName.us.oracle.com:8080`

4. At the end of `httpd.conf`, add the following lines:

   ```
   # Include mod_jk's specific configuration file
   Include conf/mod_jk.conf
   ```

5. Save the file `httpd.conf`.

13.10.3.3 Creating the File mod_jk.conf

To create the file `mod_jk.conf`, perform the following steps:

1. On Machine #1, create a new text file called `mod_jk.conf`.
2. Add the following lines to `mod_jk.conf`:

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

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

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

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

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

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

   # JkRequestLogFormat
   JkRequestLogFormat "%w %V %T"

   # Mount your applications
   JkMount /application/* loadbalancer

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

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

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

3. Save the file to *APACHE_HOME_DIR*\conf\mod_jk.conf.

   For example: C:\Program Files\Apache Software Foundation\Apache2.2\conf\mod_jk.conf

13.10.3.4 Creating the File uriworkermap.properties

To create the file uriworkermap.properties, perform the following steps:

1. On Machine #1, create a new text file called uriworkermap.properties.
2. Add the following lines to uriworkermap.properties:

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

/ui=loadbalancer
/ui/*=loadbalancer
/schema=loadbalancer
/schema/*=loadbalancer
/soap=loadbalancer
/soap/*=loadbalancer
/rtis=loadbalancer
/rtis/*=loadbalancer
/ls=loadbalancer
/ls/*=loadbalancer
```

3. Save the file to *APACHE_HOME_DIR*\conf\uriworkermap.properties.

   For example: C:\Program Files\Apache Software Foundation\Apache2.2\conf\uriworkermap.properties

13.10.3.5 Creating the File workers.properties

To create the file workers.properties, perform the following steps:

1. On Machine #1, create a new text file called workers.properties.
2. Add the following lines to workers.properties, after making the two substitutions:

   a. Replace *MACHINE_1_HOST* with the Machine #1 host URL or IP address, for example, machine1.mydomain.com

   b. Replace *MACHINE_2_HOST* with the Machine #2 host URL or IP address, for example, machine2.mydomain.com

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

# Define Node1
# modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=MACHINE_1_HOST
worker.node1.type=ajp13
worker.node1.lbfactor=1

# Define Node2
# modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=MACHINE_2_HOST
worker.node2.type=ajp13
worker.node2.lbfactor=1

# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1
#worker.list=loadbalancer

# Status worker for managing load balancer
worker.status.type=status
```

3. Save the file to `APACHE_HOME_DIR\conf\workers.properties`.

   For example: `C:\Program Files\Apache Software Foundation\Apache2.2\conf\workers.properties`

13.10.3.6 Starting the Apache HTTP Server

If the Apache Server has already been started, you can skip this section.

Before you start the Apache HTTP Server, check that the Apache 2.2 service has been installed. If it has not been installed, run the following command from the directory `C:\Program Files\Apache Software Foundation\Apache2.2\bin`:

- `httpd -k install -n "Apache2.2"`

To start the Apache HTTP Server on Machine #1, perform either of the following tasks:

- Navigate the menu path: `Start > All Programs > Apache HTTP Server 2.2.6 > Control Apache Server > Start`
- From the directory `C:\Program Files\Apache Software Foundation\Apache2.2\bin`, run the command: `httpd -k start`

The logs will be in `C:\Program Files\Apache Software Foundation\Apache2.2\logs\`.

> **Note:** To stop the Apache HTTP Server, perform either of the following:
>
> - Navigate the menu path: `Start > All Programs > Apache HTTP Server 2.2.6 > Control Apache Server > Stop`
> - From the directory `C:\Program Files\Apache Software Foundation\Apache2.2\bin`, run the command: `httpd -k stop`

13.10.4 Installing the JBoss Software

To install the JBoss software, perform the following steps:

1. On Machine #1, install JBoss EAP 4.3.

   The directory into which you installed JBoss will be referred to as *`JBOSS_HOME`*.

   For example, `C:\Program Files\EnterprisePlatform-4.3.0.GA_CP03`, `C:\jboss-eap-4.3`, or `d:\genesis\jboss-eap-4.3`

2. Repeat step 1 for Machine #2 in the cluster.

13.10.5 Configuring the JBoss Software

To configure the JBoss software, you must perform essentially the same operations on both Machine #1 and Machine #2, as described in the following sections.

13.10.5.1 Enabling the JBoss Default User

1. On Machine #1, open the file *JBOSS_HOME*/jboss-as/server/all/conf/props/jmx-console-users.properties.
2. Uncomment the sample user/password (if needed): #admin=admin.

   You can create any user name and password, for example: charles=pswd.
3. Save the file.
4. Repeat steps 1-3 on Machine #2.

13.10.5.2 Setting JBoss HTTP and HTTPS Ports

1. On Machine #1, open the file *JBOSS_HOME*/jboss-as/server/all/deploy/jboss-web.deployer/server.xml.
2. For **HTTP**, near the top of the file at "<Connector port="8080" ...", change the port to 8081, and add the following line after the "enableLookups" line:

   ```
   URIEncoding="UTF-8"
   ```

   For example:

   ```
   <Connector port="8081" address="${jboss.bind.address}"
    maxThreads="250" maxHttpHeaderSize="8192"
    emptySessionPath="true" protocol="HTTP/1.1"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    URIEncoding="UTF-8"
    connectionTimeout="20000" disableUploadTimeout="true" />
   ```
3. For **HTTPS**, near the top of the file, replace:

   ```
   <!--
   <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
      maxThreads="150" scheme="https" secure="true"
      clientAuth="false" sslProtocol="TLS" />
   -->
   ```

   with:

   ```
   <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
      maxThreads="150" scheme="https" secure="true"
      clientAuth="false" sslProtocol="TLS"
      minSpareThreads="5" maxSpareThreads="15"
      address="${jboss.bind.address}"
   keystoreFile="${rtd.home.dir}/etc/ssl/sdserver.keystore"
      URIEncoding="UTF-8"
      keystorePass="tc-ssl" />
   ```
4. Save the file.
5. Repeat steps 1-4 on Machine #2.

13.10.5.3 Setting JBoss Web Server to User Java 1.5 JVM

1. On Machine #1, open *JBOSS_HOME*/jboss-as/server/all/deploy/jboss-web.deployer/conf/web.xml.

2. Locate:

   ```
   <servlet>
      <servlet-name>jsp</servlet-name>
   ```

   and add the following servlet init parameters, if they do not already exist in the file:

   ```
   <init-param>
       <param-name>compilerSourceVM</param-name>
       <param-value>1.5</param-value>
   </init-param>
   <init-param>
       <param-name>compilerTargetVM</param-name>
       <param-value>1.5</param-value>
   </init-param>
   ```

3. Save the file.

4. Repeat steps 1-3 on Machine #2.

13.10.5.4 Adding Oracle RTD Logging to JBoss Server

1. On Machine #1, open *JBOSS_HOME*/jboss-as/server/all/conf/jboss-log4j.xml.

2. Locate the section starting with the following lines:

   ```
   <!-- ============================== -->
   <!-- Append messages to the console -->
   <!-- ============================== -->
   ```

   and add the following at the end of the above section:

   ```
   <!-- ============================================== -->
   <!-- START: Oracle RTD Appender Message Definitions -->
   <!-- ============================================== -->

   <appender name="SIGMA"
      class="com.sigmadynamics.util.SDRollingZipFileAppender">
      <param name="Threshold" value="DEBUG"/>
      <param name="File" value="${rtd.log.file}"/>
      <param name="Append" value="true"/>
      <param name="MaxFileSize" value="20000KB"/>
      <layout class="org.apache.log4j.PatternLayout">
         <param name="ConversionPattern" value="%d{ISO8601} %-5p [%c{1}] %m%n"/>
         <!-- <param name="ConversionPattern" value="%d{{HH:mm:ss,SSS}} [%t] %-5p
         [%c{1}] %m%n"/> -->
      </layout>
      <filter class="org.jboss.logging.filter.TCLFilter">
         <param name="AcceptOnMatch" value="true"/>
         <param name="DeployURL" value="RTD.ear"/>
      </filter>
      <filter class="org.apache.log4j.varia.DenyAllFilter"/>
   </appender>

   <appender name="ENG_TRACE_FILE"
   ```

```
   class="com.sigmadynamics.util.SDRollingZipFileAppender">
   <param name="Threshold" value="TRACE"/>
   <param name="File" value="${rtd.log.file}.trace"/>
   <param name="Append" value="true"/>
   <param name="MaxFileSize" value="20000KB"/>
   <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d{{HH:mm:ss,SSS}} [%t] %m%n"/>
      <!-- <param name="ConversionPattern" value="%d{{HH:mm:ss,SSS}} [%t] %-5p
      [%c{1}] %m%n"/> -->
   </layout>
</appender>

<!-- ============================================== -->
<!-- END: Oracle RTD Appender Message Definitions   -->
<!-- ============================================== -->
```

3. Locate the section starting with the following lines:

```
<!-- ====================== -->
<!-- More Appender examples -->
<!-- ====================== -->
```

and add the following at the end of the above section:

```
<!-- ============================================== -->
<!-- START: Oracle RTD Limit Category Definitions   -->
<!-- ============================================== -->

<!--
Messages logged to this TRACE category go only to the trace file, since its
additivity flag is false. This is only used temporarily by RTD engineers for
debugging low-level stuff like request queuing and load shunting. It's a place
to direct topical information while avoiding all the stuff that's going to the
normal log file.
-->

<category name="ENG_TRACE" additivity="false">
   <priority value="DEBUG"/>
   <appender-ref ref="ENG_TRACE_FILE"/>
</category>

<category name="com.sigmadynamics.server.SDDistributedHashtable">
   <priority value="ERROR"/>
</category>

<category name="com.sigmadynamics.client">
   <priority value="ERROR"/>
</category>

<category name="com.sigmadynamics">
   <priority value="DEBUG"/>
   <!-- <priority value="TRACE#com.sigmadynamics.util.XLevel"/> -->
</category>

<category name="org.jboss.messaging.core.impl.JDBCSupport">
   <priority value="INFO"/>
</category>

<category name="org.jboss.jms.server.plugin.JDBCJMSUserManager">
   <priority value="INFO"/>
</category>
```

```
<category name="org.jgroups.protocols">
   <priority value="ERROR"/>
</category>

<category name="sigmadynamics.application">
   <priority value="DEBUG"/>
</category>

<category name="request.trace.sigmadynamics.application">
   <priority value="TRACE#com.sigmadynamics.util.XLevel"/>
</category>

<category name="request.log.sigmadynamics.application">
   <priority value="DEBUG"/>
</category>

<!-- ============================================ -->
<!-- END: Oracle RTD Limit Category Definitions     -->
<!-- ============================================ -->
```

4. Locate:

   ```
   <!-- ====================== -->
   <!-- Setup the Root category -->
   <!-- ====================== -->
   ```

   and add the following line to `<root>`:

   ```
   <appender-ref ref="SIGMA"/>
   ```

5. Save the file.
6. Repeat steps 1-5 on Machine #2.

13.10.5.5 Copying Oracle RTD log4j Jar to JBoss

> **Note:** This adds Oracle RTD classes for displaying Oracle RTD server logs.

1. On Machine #1, copy *RTD_HOME*/package/jboss/rtdlog4j.jar to *JBOSS_HOME*/jboss-as/server/all/lib/.
2. Repeat step 1 on Machine #2 by copying over rtdlog4j.jar to Machine #2.

13.10.5.6 Copying New Version of the Eclipse JDT Compiler to JBoss Web Server Directory

> **Note:** This overrides the older version of the Eclipse JDT compiler jsaper-jdt.jar.
>
> See JBoss Bug #JBPAPP-869: https://jira.jboss.org/jira/browse/JBPAPP-869.

1. On Machine #1, copy *RTD_HOME*/package/jboss/ecj-3.4.jar to *JBOSS_HOME*/jboss-as/server/all/deploy/jboss-web.deployer/.

2. Repeat step 1 on Machine #2 by copying over `ecj-3.4.jar` to Machine #2.

13.10.5.7 Copying Oracle RTD jgroups Jar to JBoss

> **Note:** This replaces JBoss jgroups version 2.4 with Oracle RTD jgroups version 2.6.1.

1. On Machine #1, rename the file *`JBOSS_HOME`*`/jboss-as/server/all/lib/jgroups.jar` to `jgroups.jar.original`.
2. Copy *`RTD_HOME`*`/lib/jgroups-all2.6.1.jar` to the directory *`JBOSS_HOME`*`/jboss-as/server/all/lib/`, and rename the file in the receiving directory to jgroups.jar.
3. Repeat steps 1-2 on Machine #2 by copying over `jgroups-all2.6.1.jar` to Machine #2.

13.10.5.8 Configuring JBoss to Add Identifiers to HTTP Session IDs

1. On Machine #1, open *`JBOSS_HOME`*`/jboss-as/server/all/deploy/jboss-web.deployer/META-INF/jboss-service.xml`.
2. Locate:

   ```
   <attribute name="UseJK">
   ```

   and set the value to true:

   ```
   <attribute name="UseJK">true</attribute>
   ```

3. Save the file.
4. Repeat steps 1-3 on Machine #2.

13.10.5.9 Giving Identities to JBoss Servers to Match workers.properties Entries

1. On Machine #1, open *`JBOSS_HOME`*`/jboss-as/server/all/deploy/jboss-web.deployer/server.xml`.
2. Locate:

   ```
   <Engine name="jboss.web"
   ```

   and add an attribute `jvmRoute` as follows, with value `node1` for Machine #1 (and later `node2` for Machine #2):

   ```
   <Engine name="jboss.web"
     defaultHost="localhost" jvmRoute="node1">
     ......
   </Engine>
   ```

3. Save the file.
4. Repeat steps 1-3 on Machine #2.

13.10.5.10 Setting Up Oracle RTD SDDS Data Source and Copying to JBoss Deploy Directory

1. On Machine #1, in *RTD_HOME*/package/jboss, choose your supported database and open the file appropriate to that database, as follows:
   - For DB2: rtd-db2-ds.xml
   - For Oracle: rtd-oracle-ds.xml
   - For SQL Server: rtd-sqlserver-ds.xml
2. Replace the following:
   - ${DB_SERVER} with your database server

     For example: localhost

     > **Note:** If the database is on a SQL Server named instance, specify the name of your database server using the format *host_name\instance_name*.

   - ${DB_PORT} with your database port

     For example: 50000 for db2, 1521 for oracle, 1433 for sqlserver
   - ${DB_NAME} with your database name

     For example: RTD for db2 and sqlserver, orcl for oracle
   - ${DB_USER} with your database user name

     For example: jsmith
   - ${DB_PASSWORD} with your database user password
3. Save the file and copy it to *JBOSS_HOME*/jboss-as/server/all/deploy/.
4. Repeat step 3 on Machine #2 by copying over the appropriate "-ds.xml" file to Machine #2.

13.10.5.11 Modifying JBoss Start Script

On Windows

1. On Machine #1, open *JBOSS_HOME*/jboss-as/bin/run.bat.
2. If you modified run.bat by applying the changes described in Section 6.1.4.1, "Modifying the Windows JBoss Start Script" in *Oracle Real-Time Decisions Installation and Administration Guide*, then revert those changes.
3. Near the top of the file, add the following "set" lines, after making these substitutions:
   - Replace *RTD_HOME_TO_REPLACE* with your actual value, for example, C:\OracleBI\RTD
   - Replace *MACHINE_1_IP_ADDR* with Machine #1 IP address, for example, 10.234.7.2
   - Replace *MACHINE_2_IP_ADDR* with Machine #2 IP address, for example, 10.234.7.3

Notes: 1. You may need to replace some of the cluster-specific entries for your cluster configuration, such as `-DJGroupsMulticastPort` and `-DJGroupsMulticastAddress`. For more information on the configuration properties, see Chapter 13.3, "Cluster-Specific Configuration Properties" of *Oracle Real-Time Decisions Installation and Administration Guide*.

2. To make the cluster unique, that is, to create an independent JBoss cluster on the network, change the values of the parameters `RTD_JGROUPS_MULTICAST_ADDR` and `SDGroupName`.

   For example, replace the equivalent lines in the following script with:

```
set RTD_JGROUPS_MULTICAST_ADDR=228.64.16.55
set JAVA_OPTS=%JAVA_OPTS% -DSDGroupName=OracleRtdCluster2
```

```
set RTD_HOME=RTD_HOME_TO_REPLACE
set RTD_CLUSTER_MEMBERS=MACHINE_1_IP_ADDR;MACHINE_2_IP_ADDR
set RTD_JGROUPS_MULTICAST_ADDR=228.64.16.54

set JAVA_OPTS=-Djboss.partition.name=OracleRtdPartition
set JAVA_OPTS=%JAVA_OPTS% -Dorg.jboss.net.protocol.file.decodeFilePaths=true
set JAVA_OPTS=%JAVA_OPTS% -Djboss.partition.udpGroup=%RTD_JGROUPS_MULTICAST_
ADDR%
set JAVA_OPTS=%JAVA_OPTS% -Djboss.hapartition.mcast_port=45501
set JAVA_OPTS=%JAVA_OPTS% -Dhibernate.connection.release_mode=auto
set JAVA_OPTS=%JAVA_OPTS% -Dbind.address=localhost

set JAVA_OPTS=%JAVA_OPTS% -Dnologging=true
set JAVA_OPTS=%JAVA_OPTS% -DSDLoggingPriority=DEBUG
set JAVA_OPTS=%JAVA_OPTS% -DJGroupsMulticastAddress=%RTD_JGROUPS_MULTICAST_
ADDR%
set JAVA_OPTS=%JAVA_OPTS% -DJGroupsMulticastPort=45502
set JAVA_OPTS=%JAVA_OPTS% -DJGroupsDSMulticastAddress=%RTD_JGROUPS_MULTICAST_
ADDR%
set JAVA_OPTS=%JAVA_OPTS% -DJGroupsDSMulticastPort=45503
set JAVA_OPTS=%JAVA_OPTS% -DSDGroupName=OracleRtdCluster

set JAVA_OPTS=%JAVA_OPTS% -DTrustedClusterMembers=%RTD_CLUSTER_MEMBERS%

set JAVA_OPTS=%JAVA_OPTS% -Drtd.home.dir=%RTD_HOME%
set JAVA_OPTS=%JAVA_OPTS% -Drtd.log.file=%RTD_HOME%/log/server.log
set JAVA_OPTS=%JAVA_OPTS% -DSDLoggingFileName=%RTD_HOME%/log/server.log
set JAVA_OPTS=%JAVA_OPTS% -DDSPerfCounterLogFile=%RTD_HOME%/log/ds_perf.cvs

set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote=true
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.port=12345
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.authenticate=false
set JAVA_OPTS=%JAVA_OPTS% -Dcom.sun.management.jmxremote.ssl=false

set JAVA_OPTS=%JAVA_OPTS% -Djava.net.preferIPv4Stack=true

set JBOSS_CLASSPATH=%RTD_HOME%/lib/jdbc/bijdbc14.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/sqljdbc.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/db2jcc_license_cu.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/db2jcc.jar
set JBOSS_CLASSPATH=%JBOSS_CLASSPATH%;%RTD_HOME%/lib/jdbc/ojdbc14.jar
```

4. Save the file.
5. Repeat steps 1-3 on Machine #2.

On Unix

1. On Machine #1, open *JBOSS_HOME*/jboss-as/bin/run.sh.
2. If you modified run.sh by applying the changes described in Section 6.1.4.2, "Modifying the Unix JBoss Start Script" in *Oracle Real-Time Decisions Installation and Administration Guide*, then revert those changes.
3. Near the top of the file, add the following "set" lines, after making these substitutions:
   - Replace *JAVA_HOME_TO_REPLACE* with your actual value
   - Replace *RTD_HOME_TO_REPLACE* with your actual value, for example, C:\OracleBI\RTD
   - Replace *MACHINE_1_IP_ADDR* with Machine #1 IP address, for example, 10.234.7.2
   - Replace *MACHINE_2_IP_ADDR* with Machine #2 IP address, for example, 10.234.7.3

Notes: 1. You may need to replace some of the cluster-specific entries for your cluster configuration, such as -DJGroupsMulticastPort and -DJGroupsMulticastAddress. For more information on the configuration properties, see Chapter 13.3, "Cluster-Specific Configuration Properties" of *Oracle Real-Time Decisions Installation and Administration Guide*.

2. To make the cluster unique, that is, to create an independent JBoss cluster on the network, change the values of the parameters RTD_JGROUPS_MULTICAST_ADDR and SDGroupName.

For example, replace the equivalent lines in the following script with:

```
RTD_JGROUPS_MULTICAST_ADDR=228.64.16.55
JAVA_OPTS="$JAVA_OPTS -DSDGroupName=OracleRtdCluster2"
```

```
JAVA_HOME="JAVAHOME_TO_REPLACE"
export JAVA_HOME

RTD_HOME="RTD_HOME_TO_REPLACE"
RTD_CLUSTER_MEMBERS=MACHINE_1_IP_ADDR;MACHINE_2_IP_ADDR
RTD_JGROUPS_MULTICAST_ADDR=228.64.16.54

JAVA_OPTS=-Djboss.partition.name=OracleRtdPartition
JAVA_OPTS="$JAVA_OPTS -Dorg.jboss.net.protocol.file.decodeFilePaths=true"
JAVA_OPTS="$JAVA_OPTS -Djboss.partition.udpGroup=$RTD_JGROUPS_MULTICAST_ADDR"
JAVA_OPTS="$JAVA_OPTS -Djboss.hapartition.mcast_port=45501"
JAVA_OPTS="$JAVA_OPTS -Dhibernate.connection.release_mode=auto"
JAVA_OPTS="$JAVA_OPTS -Dbind.address=localhost"

JAVA_OPTS="$JAVA_OPTS -Dnologging=true"
JAVA_OPTS="$JAVA_OPTS -DSDLoggingPriority=DEBUG"
JAVA_OPTS="$JAVA_OPTS -DJGroupsMulticastAddress=$RTD_JGROUPS_MULTICAST_ADDR"
JAVA_OPTS="$JAVA_OPTS -DJGroupsMulticastPort=45502"
JAVA_OPTS="$JAVA_OPTS -DJGroupsDSMulticastAddress=$RTD_JGROUPS_MULTICAST_ADDR"
JAVA_OPTS="$JAVA_OPTS -DJGroupsDSMulticastPort=45503"
JAVA_OPTS="$JAVA_OPTS -DSDGroupName=OracleRtdCluster"

JAVA_OPTS="$JAVA_OPTS -DTrustedClusterMembers=$RTD_CLUSTER_MEMBERS"

JAVA_OPTS="$JAVA_OPTS -Drtd.home.dir=$RTD_HOME"
```

```
JAVA_OPTS="$JAVA_OPTS -Drtd.log.file=$RTD_HOME/log/server.log"
JAVA_OPTS="$JAVA_OPTS -DSDLoggingFileName=$RTD_HOME/log/server.log"
JAVA_OPTS="$JAVA_OPTS -DDSPerfCounterLogFile=$RTD_HOME/log/ds_perf.cvs"

JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote=true"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.port=12345"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="$JAVA_OPTS -Dcom.sun.management.jmxremote.ssl=false"

JAVA_OPTS="$JAVA_OPTS -Djava.net.preferIPv4Stack=true"

JBOSS_CLASSPATH="$RTD_HOME/lib/jdbc/bijdbc14.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/sqljdbc.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/db2jcc_license_cu.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/db2jcc.jar"
JBOSS_CLASSPATH="$JBOSS_CLASSPATH:$RTD_HOME/lib/jdbc/ojdbc14.jar"
```

4. Save the file.
5. Repeat steps 1-3 on Machine #2.

13.10.5.12 Changing Message Server Server Id on Machine #2

1. On Machine #2, open *JBOSS_HOME*/jboss-as/server/all/deploy/jboss-messaging.sar/messaging-service.xml.
2. Locate the ServerPeerID attribute line, and change the id from 0 to 1 as follows:

```
<attribute name = "ServerPeerId>${jboss.messaging.ServerPeerId:1}</attribute>
```

3. Save the file.

> **Note:** See JBoss Bug #JBAS-3968 https://jira.jboss.org/jira/browse/JBAS-3968.

13.10.6 Creating Oracle RTD Roles and Users

Before proceeding, create Oracle RTD roles and users in JBoss for each machine in your cluster. See Section 6.3, "Creating Oracle RTD Roles and Users" in *Oracle Real-Time Decisions Installation and Administration Guide* for more information.

As you follow the instructions in this section, be sure to edit the users.properties file in the all directory rather than the default directory. Specifically, the directory *JBOSS_HOME*/jboss-as/server/default/conf/users.properties that is mentioned in Section 6.3.1 in *Oracle Real-Time Decisions Installation and Administration Guide* should be *JBOSS_HOME*/jboss-as/server/all/conf/users.properties when you are using a clustered environment. Create the file if it does not already exist.

Similarly, you should edit the roles.properties file in the all directory rather than the default directory. In particular, the directory *JBOSS_HOME*/jboss-as/server/default/conf/roles.properties that is mentioned in Section 6.3.2 in *Oracle Real-Time Decisions Installation and Administration Guide* should be *JBOSS_HOME*/jboss-as/server/all/conf/roles.properties when you are using a clustered environment. Create the file if it does not already exist.

13.10.7 Starting the JBoss Application Server

To start the JBoss application server, perform the following tasks:

1. On Machine #1, open a command window (cmd) and 'cd' to: `JBOSS_HOME/jboss-as/bin/`.
2. Start JBoss by entering:

   (On Windows) `run.bat --configuration=all --host=0.0.0.0`

   (On Unix) `run.sh --configuration=all --host=0.0.0.0`
3. Repeat steps 1-2 for Machine #2.
4. Verify the identity of the IP addresses of Machine #1 and Machine #2 in their respective server.log files, located in `JBOSS_HOME\jboss-as\server\all\log\server.log`.

   For example, search for "I am" in the log file, and the IP address appears in an INFO message.

   If you have installed the Microsoft loopback adapter, the log may show a special IP address, such as 10.10.10.10. You must disable the Microsoft loopback adapter, to ensure that clustering operates as configured.

13.10.8 Deploying RTD.ear to the JBoss Cluster

To deploy `RTD.ear` to the JBoss cluster, perform the following tasks:

1. Make sure that each JBoss Server in the cluster is running.
2. On Machine #1, copy `RTD_HOME/package/RTD.ear` to `JBOSS_HOME/jboss-as/server/all/farm/`.
3. Wait for `RTD.ear` to be deployed to Machine #2.

13.10.9 Undeploying RTD.ear from the JBoss Cluster

To undeploy `RTD.ear` from the JBoss cluster, perform the following tasks:

1. Make sure that each JBoss Server in the cluster is running.
2. On Machine #1, delete `RTD.ear` from `JBOSS_HOME/jboss-as/server/all/farm/`.
3. Wait for `RTD.ear` to be undeployed from Machine #2.

13.10.10 Stopping the JBoss Application Server

To stop the JBoss application server, perform the following tasks:

1. On Machine #1, open a command window (cmd) and 'cd' to: `JBOSS_HOME/jboss-as/bin/`.
2. Stop JBoss by entering:

   (On Windows) `shutdown.bat --server=localhost:1099 -S -u` *username* `-p` *password*

   (On Unix) `shutdown.sh --server=localhost:1099 -S -u` *username* `-p` *password*

13.10.11 JBoss and Oracle RTD Logs

The JBoss log is located at `JBOSS_HOME/jboss-as/server/all/log/server.log`.

The Oracle RTD log is located at `RTD_HOME/log/server.log`.

14

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 JConsole. 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 14.1, "Setting Performance Monitoring Parameters"
- Section 14.2, "Viewing Common Performance Monitoring Snapshot Values"
- Section 14.3, "CSV File Contents"
- Section 14.4, "XLS File Contents"

14.1 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 15, "Managing Oracle Real-Time Decisions" for more information.

Table 14–1 describes the properties governing performance monitoring.

Table 14–1 Performance Monitoring Properties

Property Name	Description
DSPerfCounterEnabled	Enables the writing of DS performance counters. This property should not be enabled indefinitely, because the file grows without limit.
DSPerfCounterAppend	If true, performance data is appended to an existing file, if any, otherwise any existing file is overwritten when the server restarts.

Table 14–1 (Cont.) Performance Monitoring Properties

Property Name	Description
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.

14.2 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.

14.3 CSV File Contents

This section describes the fields of the CSV file containing performance counters.

Table 14–2 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 14–2 (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 14–2 (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 14–2 (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 14–2 (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.
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.

14.4 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.

15 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. You can use JConsole, a JMX management tool that comes with Java Development Kit 1.5.0.8+ or 1.6, to manage Oracle RTD. Third-party JMX implementations may also be used to manage Oracle RTD. This document assumes you are using JConsole.

> **Note:** Some of the management properties available through JConsole can also be set as system properties. For details, see Section 15.9, "System Properties."

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 is comprised of three Services:

Decision Center Service: Supports the deployment of Inline Services by Decision Studio. It also provides a web interface, Decision Center, for displaying the structure and decisioning history of Inline Services..

Decision Service: Runs Inline Services and integrates to enterprise operational processes.

Learning Service: Maintains analytic, self-learning models that underlie Inline Services.

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 15.1, "Accessing JConsole"
- Section 15.2, "About JMX MBean Operations and Attributes"
- Section 15.3, "Oracle Real-Time Decisions Cluster-Level Management"
- Section 15.4, "Oracle Real-Time Decisions Member-Level Management"
- Section 15.5, "Oracle Real-Time Decisions Security Management"
- Section 15.6, "Managing Inline Services"
- Section 15.7, "Deployment States"

- Section 15.8, "Managing the Learning Service"
- Section 15.9, "System Properties"

15.1 Accessing JConsole

The steps to access JConsole depend on which application server you are using.

This section contains the following topics:

- Section 15.1.1, "Accessing JConsole for OC4J, Oracle Application Server, or WebLogic"
- Section 15.1.2, "Accessing JConsole for WebSphere"

15.1.1 Accessing JConsole for OC4J, Oracle Application Server, or WebLogic

To access JConsole for OC4J, Oracle Application Server, or WebLogic:

1. Run *`JDK_HOME`*`/bin/jconsole`.

   For example:

   ```
   C:\Program Files\Java\jdk1.5.0_08\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. Click the **Remote** tab.
3. For **Host**, enter the name of the computer where Real-Time Decision Server is running.
4. For **Port**, enter your JMX remote port number (for example, `12345`).
5. For **User Name** and **Password**, enter the credentials you specified when you set up JConsole. See Section 3.9, "Setting Up JConsole Security for OC4J" and Section 5.14, "Setting Up JConsole Security for WebLogic" for more information.

15.1.2 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 4.12.2, "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`). See Section 4.12.3, "Creating a JConsole User" for more information.

   > **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.

15.2 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.

Attributes and operations at the SDCluster level are meant to manage cluster-level features. The organization of the MBeans is hierarchical. The attributes of SDCluster are:

- Properties – SD Cluster Properties Configuration.
- Members – Members of the cluster. Each Member is listed through this attribute. Member is used to manage local server level properties.
- Security – Security allows management of various authentication attributes. Many of these are specific to one authentication type and will not appear for other authentication types.
- InlineServiceManager – Manages deployed Inline Services.
- DeploymentStates – Allows the setup and ordering of deployment states.
- LearningService – Administers the Learning Service attributes.
- BatchManager – Administers the Batrch Manager Service attributes.

15.3 Oracle Real-Time Decisions Cluster-Level Management

Management at the cluster level is for items that impact the entire cluster of servers. Note that if you have only one server, there is still cluster-level management.

This section contains the following topics:

- Section 15.3.1, "About OracleRTD > SDManagement > SDClusterPropertyManager"
- Section 15.3.2, "About OracleRTD > SDClusterPropertyManager > Misc"
- Section 15.3.3, "About OracleRTD > SDClusterPropertyManager > Cluster"
- Section 15.3.4, "About OracleRTD > SDClusterPropertyManager > Deployment"
- Section 15.3.5, "About OracleRTD > SDCluster > SDManagement"

15.3.1 About OracleRTD > SDManagement > SDClusterPropertyManager

The SDManagement > SDClusterPropertyManager MBean has the following attributes:

Attribute	Description
Cluster	Cluster configuration.
Deployment	Configuration for deployment.
Misc	Miscellaneous properties.

The SDManagement > SDClusterPropertyManager MBean has the following operation:

java.lang.Void restoreDefault()

Restores the default installation settings to Oracle RTD. If this command is run on a cluster, then values are restored to cluster defaults.

15.3.2 About OracleRTD > SDClusterPropertyManager > Misc

The SDClusterPropertyManager > Misc MBean has the following attributes:

Attribute	Description
ArchivedModelCacheTimeToLive	Maximum time in seconds an archived model is preserved in memory. The cache of archived models is used by Discovery Explorer.
ArchivedModelCatalogRefreshInterval	Refresh interval in seconds for a catalog of archived models. The catalog is used by Discovery Explorer.
AutoFlushTimeout	Interval in seconds controlling auto flush of database write buffers. Fractional values are supported.
DBOperationLogThresholdMilliSec	All database operations that take longer than the specified threshold are logged.
DCOperationLogThresholdMilliSec	All decision center requests that take longer than the specified threshold are logged.
DSEnforcesStrictSessionAffinity	When set to true, and DSManagesSessionAffinity is also true, the decision service instances assure that updates to the cluster's session key map are made transactionally, to avoid the possibility of two sessions existing on different hosts with common keys. When set to false, it is possible for sessions to exist on separate hosts with common keys if the initial requests arrive at the load balancer in close proximity.
DSManagesSessionAffinity	Decision Service manages session affinity. When set to true, the decision service maintains a map of active session keys and, if necessary, will forward Integration Point requests to the cluster host owning the key's session. Should be disabled in single-host installations and in installations where session affinity is perfectly managed by the application server or external load balancer.
DSSessionIdleTimeoutMillisec	Decision Service session idle timeout in milliseconds.
DSStrictSessionAffinityConcurrency	How many concurrent DB requests can be outstanding, per Oracle RTD instance, to manage the strict Decision Service session directory. Keep this low for SQL Server to avoid DB-centric deadlocks. 0 means unlimited.
DatabaseComponentCloseTimeoutSeconds	How long to wait for the database provider to close an individual component (for example, each BatchUpdater) before abandoning the effort.
DatabaseShutdownTimeoutSeconds	How long to wait for the database provider to shutdown all its components before abandoning the effort.
DecisionServiceAddress	Must be set for Decision Center to be able to test integration point requests from its Interactive Integration Map when Decision Center is not co-located with Decision Service.
DisableBatchDBOperations	Boolean switch that controls batch database operations.

Attribute	Description
IntegrationPointGuaranteed RequestTimeout	Guaranteed response time, in milliseconds, for Integration Point requests. (Service Level Guarantee). Zero means don't timeout Integration Point requests - suitable for debugging only.
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. The calculated value of IntegrationPointMaxConcurrentJobs can be seen in SDConsole/Members/Decision Service.
IntegrationPointRequestIOFactor	The percentage of time Integration Point requests spend doing IO, or otherwise waiting for systems external to this VM.
ModelDSName	The JNDI name of the datasource used by the Learning Service.
ModelSnapshotDSName	The JNDI name of the datasource for the model snapshots.
ModelSnapshotMinAbsCorrelation	Controls whether to dump all correlation rows or to set a minimum correlation value for dumping. 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.
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. 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.
SystemDSName	The JNDI name of the datasource.
WorkerThreadPoolSize	The number of threads used for general purpose maintenance activities, not for normal Integration Point request processing. Maintenance activities include model maintenance, session timing, and timed-out request processing.

15.3.3 About OracleRTD > SDClusterPropertyManager > Cluster

The SDClusterPropertyManager > Cluster MBean has the following attributes:

Attribute	Description
BatchManagerInitialWait	The number of milliseconds to wait when the server first starts up before trying to start the Batch Manager Service.

Attribute	Description
BatchManagerRestartWait	The number of milliseconds to wait after a computer fails or leaves the cluster before trying to restart the Batch Manager Service.
ChoiceHistoryCleanupChunkSize	The chunk size to use when deleting old choice history records.
ChoiceHistoryCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.
GenerateDSCookies	Generate Decision Server HTTP Cookies. Set to `true` to have Decision Server associate Integration Point requests with HTTP sessions, thus causing the Web container to generate container-specific session-affinity cookies.
GenerateDSSessionIdCookie	Generate Decision Service Session-ID Cookies. Set to true to have Decision Service supply a cookie named ORTD_DS_SessionID identifying the DS session serving the current integration point request.
JGroupsDSMulticastAddress	Gets merged into the JGroupsDSProtocols string as the multicast address used by the Decision Service forwarding channelups.
JGroupsDSMulticastPort	Gets merged into the JGroupsDSProtocols string as the multicast port for the Decision Service forwarding channel.
JGroupDSProtocols	Defines the JGroups protocols and properties used for the Decision Server forwarding channel.
JGroupsMulticastAddress	Gets merged into the JGroupsProtocols string as the multicast address used by Oracle RTD cluster management.
JGroupsMulticastPort	Gets merged into the JGroupProtocols string as the multicast port used by RTD cluster management.
JGroupProtocols	Defines the protocols and properties used for the JGroups channel.
LearningDataStorageCleanupChunkSize	The chunk size to use when deleting old learning data storage records.
LearningDataStorageCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.
LearningServiceInitialWait	The number of milliseconds to wait when the server first starts up before trying to start the Learning Service.
LearningServiceRestartWait	The number of milliseconds to wait after a computer fails or leaves the cluster before trying to restart the Learning Service.
OperationalDataCleanupPeriod	The number of hours (fractions are allowed) between cleanup of the operational data (choice history, statistics, learning data storage) in the database.
RestrictClusterMembers	True if the hosts that can be in the cluster is restricted to a fixed list of trusted IP addresses.
RestrictDSClients	True if the hosts that can send Decision Service requests is restricted to a fixed list of trusted IP addressess.
SDGroupName	The name of the Oracle RTD cluster, as recognized by Oracle RTD's JGroups cluster management.
StatisticsCleanupChunkSize	The chunk size to use when deleting old statistic records.
StatisticsCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.

Attribute	Description
TrustedClusterMembers	List of host IP addresses that can join the cluster. Port is optional, separated from IP by ':'. Entries are separated by ';'.
TrustedDSClients	List of host IP addresses from which Decision Service requests will be accepted. Port is optional, separated from IP by ':'. Entries are separated by ';'.

15.3.4 About OracleRTD > SDClusterPropertyManager > Deployment

The SDClusterPropertyManager > Deployment MBean has the following attribute:

Attribute	Description
AppPollingInterval	How frequently, in seconds, the AppFactory polls the SDApps table to see if there are new apps.

15.3.5 About OracleRTD > SDCluster > SDManagement

The SDCluster > SDManagement MBean has the following attributes:

Attribute	Description
BatchManager	Batch Manager configuration.
DeploymentStates	Manage deployment states.
InlineServiceManager:	Inline Service Manager.
LearningService	Learning Service configuration.
Members	Cluster members.
Properties	Cluster properties configuration.
Security	Security Manager.

15.4 Oracle Real-Time Decisions Member-Level Management

This section provides information about management at the member level.

This section contains the following topics:

- Section 15.4.1, "About OracleRTD > SDManagement > SDLogging"
- Section 15.4.2, "About OracleRTD > SDManagement > SDPropertyManager"
- Section 15.4.3, "About OracleRTD > SDPropertyManager > Performance Monitoring"
- Section 15.4.4, "About OracleRTD > SDPropertyManager > Misc"
- Section 15.4.5, "About OracleRTD > Server > DecisionService"
- Section 15.4.6, "About OracleRTD > Server > SDManagement"
- Section 15.4.7, "About OracleRTD > Server > BatchAgent"
- Section 15.4.8, "About OracleRTD > Server > BatchManager"

15.4.1 About OracleRTD > SDManagement > SDLogging

The SDManagement > SDLogging MBean has the following attributes:

Attribute	Description
InlineServicePriority	Current logging priority for Inline Services. Valid values are `DEBUG`, `ERROR`, `INFO`, and `WARN`.
LogFile	Location of the current log file.
Priority	Current logging priority for the `com.sigmadynamics` category. Valid values are `DEBUG`, `ERROR`, `INFO`, and `WARN`.

For more information about the values written to the log file, see:

- Appendix B, "Exceptions"
- Appendix A, "Development Error Messages" in *Oracle Real-Time Decisions Platform Developer's Guide*

15.4.2 About OracleRTD > SDManagement > SDPropertyManager

The SDManagement > SDPropertyManager MBean has the following attributes:

Attribute	Description
PerformanceMonitoring	Performance counter properties.
Misc	Miscellaneous properties.

15.4.3 About OracleRTD > SDPropertyManager > Performance Monitoring

The SDPropertyManager > Performance Monitoring MBean has the following attributes:

Attribute	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 should not be enabled indefinitely, because the file grows without limit.
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 installation's `etc` directory, provides a convenient view.
DSPerfCounterLogInterval	The update interval in milliseconds for DS performance counts.

For more information about using performance monitoring, see Chapter 14, "Performance Monitoring".

15.4.4 About OracleRTD > SDPropertyManager > Misc

The SDPropertyManager > Misc MBean has the following attributes:

Attribute	Description
BatchAgentEnabled	Whether or not Batch Agent should run in this instance.
BatchManagerEnabled	Whether or not Batch Manager should run in this instance.

Attribute	Description
DecisionCenterEnabled	Whether or not Decision Center should run in this instance.
DecisionServiceEnabled	Whether or not Decision Service should run in this instance.
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. The calculated value of IntegrationPointMaxConcurrentJobs can be seen in `SDConsole/Members/Decision Service`.
IntegrationPointRequestIOFactor	The percentage of time Integration Point requests spend doing IO, or otherwise waiting for systems external to this VM.
LearningServiceEnabled	Whether or not Learning Service should run in this instance.
WorkerThreadPoolSize	The number of threads used for general purpose maintenance activities, not for normal Integration Point request processing. Maintenance activities include model maintenance, session timing, and timed-out request processing.

15.4.5 About OracleRTD > Server > DecisionService

The Server > DecisionService MBean has the following read-only attributes:

Attribute	Description
CurrentRequestsQueued	Number of currently waiting requests, not yet running.
CurrentRequestsRunning	Number of currently running requests.
CurrentSessions	Number of Decision Service sessions still open.
MaxAllowedConcurrentRequests	Maximum number of requests that can run concurrently.
NumCPUs	Number of CPUs in this host. In Windows, for a hyper-threaded computer, use half the number of processors shown by Windows Task Manager.
PeakRequestsQueued	Largest number of requests that have had to wait at any one time.
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.

Attribute	Description
TotalSessions	Total number of Decision Service sessions created.

15.4.6 About OracleRTD > Server > SDManagement

The Server > SDManagement MBean has the following attributes:

Attribute	Description
BatchAgent	Batch Agent configuration.
DecisionService	Decision Service configuration.
Logger	Logger configuration.
Properties	Properties configuration.

15.4.7 About OracleRTD > Server > BatchAgent

The Server > BatchAgent 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.

15.4.8 About OracleRTD > Server > BatchManager

The Server > BatchManager 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.

15.5 Oracle Real-Time Decisions Security Management

Security management MBeans impact authentication settings on the cluster. Different security settings are displayed, depending on the type set in Security Authenticator.

This section contains the following topic:

- Section 15.5.1, "About OracleRTD > SDManagement > SecurityManager"

15.5.1 About OracleRTD > SDManagement > SecurityManager

The SDManagement > SecurityManager MBean has the following operations:

assignPermission()

Assigns the specified cluster permission. See Section 7.4, "Assigning Permissions" for more information. This operation accepts the following parameters:

UserOrGroup: The user name or group name to which you want to assign permissions.

permCode: The code of the permission.

listDirectPermissions()

Returns the permissions directly assigned to a specific user or group. This operation accepts the following parameter:

userOrGroup: The target user or group name.

listEveryoneHavingDirectPermissions()

Returns the names of users and groups 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 user or group. This operation accepts the following parameter:

userOrGroup: The target user or group name.

removePermission()

Removes the specified permission. This operation accepts the following parameters:

userOrRole: The target user name or role name.

permCode: The code of the permission you want to remove.

revertToStandardPermissions()

Reverts to the standard permissions.

15.6 Managing Inline Services

Use Inline Service Manager to manage the Inline Services deployed on the cluster.

This section contains the following topics:

- Section 15.6.1, "About OracleRTD > SDManagement > InlineServiceManager"
- Section 15.6.2, "About OracleRTD > InlineServiceManager > [Inline Service] > [Deployment State] > [Flag]"
- Section 15.6.3, "Invoking Maintenance Operations"

15.6.1 About OracleRTD > SDManagement > InlineServiceManager

Each deployed Inline Service is displayed under the InlineServiceManager MBean.

The SDManagement > InlineServiceManager MBean has the following attribute:

Attribute	Description
InlineServices	List of deployed inline services.

The SDManagement > InlineServiceManager MBean has the following operation:

refreshMBeans()

Removes MBeans for applications no longer in the database, and creates MBeans for new ones.

removeAllServices()

Removes all Inline Services (loaded, loadable, failed).

15.6.2 About OracleRTD > InlineServiceManager > [Inline Service] > [Deployment State] > [Flag]

InlineServiceManager 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 > InlineServiceManager > CrossSell > Production > Loadable.

The InlineServiceManager 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 InlineServiceManager 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.

15.6.3 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.

15.7 Deployment States

By viewing the OracleRTD > DeploymentStates MBeans, you can see a list of deployment states that are available on the cluster.

This section contains the following topics:

- Section 15.7.1, "About OracleRTD > SDManagement > DeploymentStates"
- Section 15.7.2, "About OracleRTD > Deployment States > [State]"

15.7.1 About OracleRTD > SDManagement > DeploymentStates

The SDManagement > DeploymentStates MBean has the following attribute:

Attribute	Description
StateObjectNames	A listing of all deployment states available on the server.

15.7.2 About OracleRTD > Deployment States > [State]

MBeans for a particular Deployment State can be viewed by choosing OracleRTD > Deployment States, then choosing a Deployment State (for example, Development, QA, or Production).

Each Deployment States > [State] MBean has the following attributes:

Attribute	Description
AllowHotSwapping	Allow hot swapping of Inline Services with this deployment state in Decision Service.
Id	ID of the deployment state.
Name	Name of the deployment state.

> **Note:** If hot swapping is enabled for a deployment state, and an Inline Service is redeployed in the state, the existing Inline Service will remain active until all existing sessions close or timeout. New sessions will be created on the newly deployed Inline Service.

15.8 Managing the Learning Service

Managing the Learning Service on the cluster allows you to check the status of the learning models and perform maintenance on them.

This section contains the following topics:

- Section 15.8.1, "About OracleRTD > Server > LearningService"
- Section 15.8.2, "About OracleRTD > Learning Server > [Study]"
- Section 15.8.3, "About OracleRTD > Study > [Study Name] > [Model]"

15.8.1 About OracleRTD > Server > LearningService

The Server > LearningService MBean has the following attribute:

Attribute	Description
Studies	A list of all Studies running on the Learning Server. The models of a Study are viewed by clicking on a Study.

15.8.2 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:

CompleteSnapshot()

Saves a snapshot of this study to the database (complete).

Delete()

Deletes this study.

DeleteSnapshot()

Removes this study's snapshot from the database.

IncrementalSnapshot()

Saves a snapshot of this study to the database (delta).

15.8.3 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.

15.9 System Properties

You can view and set the following system properties:

System Property	Description
JGroupsDSMulticastAddress	Gets merged into the JGroupsDSProtocols string as the multicast address used by the Decision Service forwarding channelups.
JGroupsDSMulticastPort	Gets merged into the JGroupsDSProtocols string as the multicast port for the Decision Service forwarding channel.
JGroupsMulticastAddress	Gets merged into the JGroupsProtocols string as the multicast address used by RTD cluster management.
JGroupsMulticastPort	Gets merged into the JGroupProtocols string as the multicast port used by RTD cluster management.
ModelDSName	The JNDI name of the datasource used by the Learning Service. Default=SDDS.
ModelSnapshotDSName	The JNDI name of the datasource used by the Learning Service to perform snapshots of its learning models. Default=SDDS.
RestrictClusterMembers	True if the hosts that can be in the cluster is restricted to a fixed list of trusted IP addresses.
RestrictDSClients	True if the hosts that can send Decision Service requests is restricted to a fixed list of trusted IP addressess.
SDGroupName	This is the name of the Oracle RTD cluster, as recognized by Oracle RTD's JGroups cluster management.
SystemDSName	This is the name of the Oracle RTD system datasource. Default=SDDS.
TrustedClusterMembers	List of host IP addresses that can join the cluster. Port is optional, separated from IP by ':'. Entries are separated by ';'.
TrustedDSClients	List of host IP addresses from which Decision Service requests will be accepted. Port is optional, separated from IP by ':'. Entries are separated by ';'.

A

System Log and Configuration Files

This appendix provides the names and locations of the Oracle RTD log and configuration files, listed by application server type. 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, "Log Files"
- Section A.2, "Configuration Files"

A.1 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.1.1, "Main Oracle RTD Log Files"
- Section A.1.2, "Log Files for Oracle RTD Client Tools"
- Section A.1.3, "Server-Side Application Server Specific Log Files"
- Section A.1.4, "Eclipse Log File"

A.1.1 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/server.log`
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**
Standalone OC4J	`OC4J_HOME/j2ee/home/application/OracleRTD/`
Oracle AS	`ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/`
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/`
JBoss	`RTD_HOME/` (for runtime `server.log` and performance log `ds_perf.cvs`) `JBOSS_HOME/jboss-as/server/default/tmp/deploy/tmp<...>RTD.ear-contents`, where <...> is a random number.

Setting Logging Levels for the Oracle RTD Server Log

To change logging options for the Oracle RTD Server log, use JConsole. See Section 15.4.1, "About OracleRTD > SDManagement > SDLogging" 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.1.2 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.1.3 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.1.3.1 Server-Side Log Files for Standalone OC4J

```
OC4J_HOME/j2ee/home/application-deployments/OracleRTD/application.log
OC4J_HOME/j2ee/home/log/server.log
OC4J_HOME/j2ee/home/log/oc4j/log.xml
```

A.1.3.2 Server-Side Log Files for Oracle Application Server

```
ORACLE_AS_HOME/j2ee/oc4j_instance/application-deployments/OracleRTD/
home_default_group_1/application.log
```

```
ORACLE_AS_HOME/j2ee/oc4j_instance/og/home_default_group_1/server.log
ORACLE_AS_HOME/j2ee/oc4j_instance/log/home_default_group_1/oc4j/log.xml
ORACLE_AS_HOME/opmn/logs/opmn.log
ORACLE_AS_HOME/opmn/logs/default_group~home~default_group~1.log
```

A.1.3.3 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.1.3.4 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/
profile_name.log
MW_HOME/user_projects/domains/domain_name/servers/server_name/logs/
server_name.out
```

A.1.3.5 Server-Side Log Files for JBoss

```
JBOSS_HOME/jboss-as/server/default/log/boot.log
JBOSS_HOME/jboss-as/server/default/log/server.log
```

The file server.log contains all the Oracle RTD log information as well as additional server log information.

A.1.4 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:\Documents and Settings\ <*windows_user_name*>\Oracle Decision Studio\.

For example, C:\Documents and Settings\jsmith\Oracle Decision 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_3.0.0\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.2 Configuration Files

This section lists the configuration files specific to the application servers supported by Oracle RTD.

Oracle RTD Configuration Files for Standalone OC4J

```
OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdconfig-cluster.xml
OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdconfig-machine.xml
OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdconfig-tmpl.xml
OC4J_HOME/j2ee/home/applications/OracleRTD/etc/sdlog4j.xml
```

Oracle RTD Configuration Files for Oracle Application Server

```
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/sdconfig-cluster.xml
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/sdconfig-machine.xml
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/sdconfig.xml
ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/sdlog4j.xml
```

Oracle RTD Configuration Files for WebSphere

```
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_cell_name/
OracleRTD.ear/etc/sdconfig-cluster.xml
WEBSPHERE_HOME/AppServer/profiles/profile_name/installedApps/host_node_cell_name/
OracleRTD.ear/etc/sdconfig-machine.xml
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-cluster.xml
MW_HOME/user_projects/domains/domain_name/servers/server_
name/stage/OracleRTD/etc/sdconfig-machine.xml
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
```

Oracle RTD Configuration Files for JBoss Single Server

Note: These files are only available while the JBoss server is running. These files are removed upon JBoss shutdown.

```
JBOSS_
HOME/jboss-as/server/default/tmp/deploy/tmp<...>RTD.ear-contents/etc/sdconfig-clus
ter.xml
JBOSS_
HOME/jboss-as/server/default/tmp/deploy/tmp<...>RTD.ear-contents/etc/sdconfig-mach
ine.xml
JBOSS_
HOME/jboss-as/server/default/tmp/deploy/tmp<...>RTD.ear-contents/etc/sdconfig.xml
JBOSS_
HOME/jboss-as/server/default/tmp/deploy/tmp<...>RTD.ear-contents/etc/sdlog4j.xml
```

where <...> is a random number.

B

Exceptions

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 15, "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.
Could not find JMX Server *server_name*	If the JMX Server is unavailable, check the database connectivity and make sure the server is running.

Table B–1 (Cont.) List of Exceptions

Exception	Action
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 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.