

# **Oracle® Real-Time Decisions**

## **Installation and Administration of Oracle RTD**

Copyright © 2003, 2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

**U.S. GOVERNMENT RIGHTS.** Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

# Installation and Administration of Oracle RTD

1	Preparing for Installation .....	9
1.1	About Oracle Real-Time Decisions installation .....	9
1.2	Installing the database server.....	10
1.3	Installing Java Development Kit (JDK) .....	10
1.3.1	Standalone OC4J Installations .....	10
1.3.2	Oracle Application Server Installations .....	11
1.3.3	WebSphere Installations.....	11
1.3.4	WebLogic Installations.....	11
1.4	Installing the application server .....	12
1.4.1	Installing OC4J for Oracle RTD .....	12
1.4.2	Installing WebSphere for Oracle RTD .....	13
1.4.3	Installing WebLogic for Oracle RTD .....	13
1.5	Additional preinstallation tasks .....	13
2	Installing the Real-Time Decision Server .....	15
2.1	Installing Oracle Real-Time Decisions files .....	15
2.1.1	Installing Oracle RTD on a single Windows computer.....	15
2.1.2	Installing Real-Time Decision Server on UNIX.....	15
2.1.3	About the Oracle RTD run-time environment .....	16
2.2	Creating and initializing the Oracle RTD Database .....	17
2.2.1	Creating a database: Microsoft SQL Server .....	17
2.2.2	Creating a database: Oracle Database .....	17
2.2.3	Creating a database: IBM DB2.....	17
2.2.4	Initializing the Oracle RTD Database using SDDBTool.....	18
2.3	Populating the CrossSell example data (optional) .....	18
2.4	Using SSL with Oracle Real-Time Decisions .....	19
3	Configuring OC4J for Oracle Real-Time Decisions.....	21

3.1	Configuring server properties .....	21
3.2	Creating a JDBC provider for the Oracle RTD Database .....	24
3.3	Installing the Oracle Real-Time Decisions application on OC4J .....	26
3.4	Starting Oracle Real-Time Decisions .....	26
3.5	Uninstalling the Oracle Real-Time Decisions application from OC4J.....	27
3.6	Configuring SSL for Real-Time Decision Server (recommended).....	27
3.6.1	Testing the SSL configuration.....	29
3.7	Setting up JConsole for OC4J .....	30
3.8	Changing the Oracle Real-Time Decisions port number .....	31
3.8.1	Changing the port number in OC4J .....	31
3.8.2	Changing the port number in Oracle RTD .....	32
4	Configuring WebSphere for Oracle Real-Time Decisions .....	33
4.1	Configuring server properties .....	33
4.2	Creating a JDBC provider for the Oracle RTD Database .....	35
4.3	Installing the Oracle Real-Time Decisions application on WebSphere .....	37
4.4	Starting Oracle Real-Time Decisions .....	38
4.5	Uninstalling the Oracle Real-Time Decisions application from WebSphere.....	38
4.6	Configuring SSL for Real-Time Decision Server (recommended).....	38
4.6.1	Testing the SSL configuration.....	41
4.7	Setting up JConsole for WebSphere .....	41
4.7.1	Determining the WebSphere bootstrap port number .....	41
4.7.2	Setting up a batch file for JConsole .....	42
4.7.3	Creating a JConsole user .....	43
4.8	Changing the Oracle Real-Time Decisions port number .....	43
4.8.1	Changing the port number in WebSphere .....	44
4.8.2	Changing the port number in Oracle RTD .....	44
5	Configuring WebLogic for Oracle Real-Time Decisions .....	45
5.1	Configuring server properties .....	45

5.2	Creating a JDBC provider for the Oracle RTD Database .....	49
5.3	Installing the Oracle Real-Time Decisions application on WebLogic .....	50
5.4	Starting Oracle Real-Time Decisions .....	50
5.5	Uninstalling the Oracle Real-Time Decisions application from WebLogic.....	51
5.6	Configuring SSL for Real-Time Decision Server (recommended).....	51
5.6.1	Testing the SSL configuration.....	53
5.7	Setting up JConsole for WebLogic .....	53
5.8	Changing the Oracle Real-Time Decisions port number.....	54
5.8.1	Changing the port number in WebLogic .....	55
5.8.2	Changing the port number in Oracle RTD .....	55
6	Configuring Authentication for Oracle Real-Time Decisions .....	56
6.1	Oracle Real-Time Decisions security overview .....	56
6.1.1	Oracle RTD default accounts and passwords .....	56
6.1.2	Oracle RTD configuration and log files .....	57
6.1.3	Oracle RTD cookies.....	59
6.2	Configuring Windows Authentication.....	60
6.3	Configuring Oracle RTD Platform Authentication .....	61
6.4	Configuring Oracle's Siebel Object Manager Authentication.....	62
6.4.1	Providing access to jar files for your application server.....	62
6.4.2	Enabling authentication in Oracle RTD.....	63
6.5	Assigning permissions to users and groups .....	64
6.6	Changing the authentication type .....	64
7	Configuring Data Access for Oracle Real-Time Decisions.....	66
7.1	Creating additional JDBC data sources in OC4J.....	66
7.1.1	Providing access to JDBC jar files for your data source.....	66
7.1.2	Creating a connection pool and data source in OC4J .....	67
7.1.3	Adding the new data source to Oracle RTD .....	68
7.2	Creating additional JDBC data sources in WebSphere.....	70

7.2.1	Creating a data provider and data source in WebSphere.....	70
7.2.2	Adding the new data source to Oracle RTD .....	72
7.2.3	Updating resource references in WebSphere.....	73
7.2.4	Adding additional data sources.....	73
7.3	Configuring additional JDBC data sources in WebLogic.....	73
7.3.1	Setting the path to JDBC jar files for your data source.....	74
7.3.2	Creating a data source in WebLogic.....	75
7.3.3	Adding the new data source to Oracle RTD .....	77
7.4	Testing a new enterprise data source .....	78
8	Additional Configuration Settings and Starting Client Tools.....	79
8.1	Decision Center Browser Configuration .....	79
8.2	Real-Time Decision Server Logging Configuration .....	79
8.3	Accessing Oracle Real-Time Decisions clients .....	79
8.3.1	Accessing Decision Studio .....	80
8.3.2	Accessing Decision Center.....	80
8.3.3	Accessing Load Generator .....	80
9	Production Deployment of Oracle Real-Time Decisions .....	81
9.1	All-In-One configuration.....	81
9.2	Full Deployment configuration .....	81
10	JMX Management of Oracle Real-Time Decisions .....	83
10.1	Accessing JConsole .....	83
10.2	About JMX MBean operations and attributes .....	84
10.3	Oracle Real-Time Decisions cluster-level management.....	85
10.3.1	About OracleRTD > SDManagement > SDClusterPropertyManager .....	85
10.3.2	About OracleRTD > SDClusterPropertyManager > Misc.....	85
10.3.3	About OracleRTD > SDClusterPropertyManager > Cluster.....	86
10.3.4	About OracleRTD > SDClusterPropertyManager > Deployment.....	87
10.3.5	About Oracle RTD > SDClusterPropertyManager > Alert Service.....	88

10.4	Oracle Real-Time Decisions member-level management .....	88
10.4.1	About OracleRTD > SDManagement > SDLogging .....	88
10.4.2	About OracleRTD > SDPropertyManager > Performance Monitoring.....	88
10.4.3	About OracleRTD > SDPropertyManager > Misc.....	89
10.4.4	About OracleRTD > Server > DecisionService.....	90
10.5	Oracle Real-Time Decisions security management .....	90
10.5.1	About OracleRTD > SDManagement > SecurityManager .....	90
10.5.2	About OracleRTD > SDClusterPropertyManager > Security Properties.....	93
10.5.3	About OracleRTD > SDManagement > SecurityAuthenticator .....	93
10.6	Managing Inline Services .....	94
10.6.1	About OracleRTD > SDManagement > InlineServiceManager .....	94
10.6.2	About OracleRTD > InlineServiceManager > [Inline Service] > [Deployment State] > [Flag] .....	94
10.7	Managing Deployment States .....	95
10.7.1	About OracleRTD > SDManagement > DeploymentStates.....	95
10.7.2	About OracleRTD > Deployment States > [State] .....	96
10.8	Managing the Learning Service.....	96
10.8.1	About OracleRTD > Server > LearningService.....	96
10.8.2	About OracleRTD > Learning Server > [Study] .....	97
10.8.3	About OracleRTD > Study > [Study Name] > [Model] .....	97
10.9	Managing the Alert Service .....	98
10.9.1	About OracleRTD > SDClusterPropertyManager > Alert Service.....	98

# 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.

## About this document

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

## Intended 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.

## How to use this guide

This document is divided into the following sections:

**Section 1: Preparing for Installation** outlines tasks that must be performed before you begin installation.

**Section 2: Installing the Real-Time Decision Server** gives step-by-step instructions on how to install files for Real-Time Decision Server and Oracle RTD clients, and how to create and initialize the Oracle RTD Database.

**Section 3: Configuring OC4J for Oracle Real-Time Decisions** provides information about how to install Oracle RTD on Oracle Containers for J2EE (OC4J).

**Section 4: Configuring WebSphere for Oracle Real-Time Decisions** provides information about how to install Oracle RTD on WebSphere.

**Section 5: Configuring WebLogic for Oracle Real-Time Decisions** provides information about how to install Oracle RTD on WebLogic.

**Section 6: Configuring Authentication for Oracle Real-Time Decisions** provides information about setting up supported authentication methods.

**Section 7: Configuring Data Access for Oracle Real-Time Decisions** explains how to set up JDBC data sources.

**Section 8: Additional Configuration Settings** provides information about additional configuration requirements for Oracle RTD clients and servers.




**Section 9: Configuring Data Access for Oracle Real-Time Decisions** discusses deployment scenarios for production environments.

**Section 10: JMX Management of Oracle Real-Time Decisions** describes the JMX MBeans used to manage Oracle RTD.

## Document conventions

Convention	Description
<code>monospace</code>	Indicates source code and program output.



Convention	Description
<b>bold</b>	Indicates portions of the user interface, such as labels, tabs, and menus.
<i>italic</i>	Indicates user-specific values.
'quote'	Indicates input required from the user.
	Indicates additional information that may make the task easier.
	Indicates additional information about the subject.
	Indicates actions that may result in loss of data or errors.

#### 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), BEA WebLogic, or IBM WebSphere application servers. For detailed information about supported platforms, refer to *System Requirements and Supported Platforms for Oracle Real-Time Decisions for Version 2.2*.

**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 section contains the following topics:

- 1.1 About Oracle Real-Time Decisions installation
- 1.2 Installing the database server
- 1.3 Installing Java Development Kit (JDK)
- 1.4 Installing the application server
- 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 *System Requirements and Supported Platforms for Oracle Real-Time Decisions* for Version 2.2.

Note the following:

- If you are using SQL Server, you must install your database server as the default (unnamed) 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.

### 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 later 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 later before proceeding with the installation of Oracle RTD. Then, install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).

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. Then, install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).

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. Then, install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).

### 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. However, you do need to install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).

### 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 **UNIX** (Solaris, Linux, HP-UX, or AIX-based systems), you do not need to download a separate JDK. WebSphere comes with its own JDK. However, you do need to install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).

**Note:** For all WebSphere installations except for Linux, you must update the JDK that comes with WebSphere by installing WebSphere Fix Pack 3. See Section 1.4.2 for more information.

### 1.3.4 WebLogic Installations

If you plan to install Real-Time Decision Server on WebLogic on **Windows**, you do not need to download a separate JDK. WebLogic 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 WebLogic.

If you plan to install Real-Time Decision Server on WebLogic on **Solaris or Linux**, you do not need to download a separate JDK. WebLogic comes with its own JDK. However, you do need to install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).

If you plan to install Real-Time Decision Server on WebLogic on **HP-UX**, you must install HP-UX JDK 5.0.05 before proceeding with the installation of Oracle RTD. Then, install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).

If you plan to install Real-Time Decision Server on WebLogic on **AIX-based systems**, you must install IBM Java 5 SR2 before proceeding with the installation of Oracle RTD. Then, install Sun JDK version 1.5.0.8 or later on the Windows computer where you plan to install the Oracle RTD client tools (such as Decision Studio, Load Generator, and SDDBTool).



**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
```

## 1.4 Installing the application server

Real-Time Decision Server runs on one of the following application servers: OC4J (a component of Oracle Application Server), WebSphere, or WebLogic. For detailed information about supported application server versions, refer to *System Requirements and Supported Platforms for Oracle Real-Time Decisions for Version 2.2*.

### 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

- 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

- 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 *WEBSHERE\_HOME*. For example, on Windows, the default *WEBSHERE\_HOME* is `C:\Program Files\IBM\WebSphere`.

- 2 Update the WebSphere installation to V6.1.0 Fix Pack 3. You can download the appropriate updates at <http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg24013830>.

For all platforms other than Linux, you must install the update for both the **AppServer** and **Java SDK** components. For Linux platforms, only install the update for the **AppServer** component; do not install the update for **Java SDK**. If you install the **Java SDK** update on Linux, the WebSphere server may not start.

#### 1.4.3 Installing 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.

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

- 3 If you installed WebLogic on Windows, when you set up the WebLogic domain you want to use for Oracle RTD, make sure to select the Sun SDK in the Configure Server Start Mode and JDK screen of the Configuration Wizard. Do not select the JRockit SDK.

Refer to the WebLogic documentation for additional information about creating WebLogic domains. See Section 1.3.4, WebLogic Installations for more information about JDK requirements for WebLogic.

- 4 If you have an existing WebLogic domain that you want to use for Oracle RTD that points to the wrong JDK, you must modify your WebLogic configuration so that the domain uses the correct JDK.

## 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 the 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 section contains the following topics:

- 2.1 Installing Oracle Real-Time Decisions files
- 2.2 Creating and initializing the Oracle RTD Database
- 2.3 Populating the CrossSell example data (optional)
- 2.4 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.

#### 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_2.2_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 the Real-Time Decision Server on a UNIX computer, and install the Oracle RTD components 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_2.2_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_win_oc4j_2.2.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 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/config/default-web-site.xml`.

- If you are using WebSphere, `RTD_RUNTIME_HOME` is `WEBSHERE_HOME/AppServer/profiles/profile_name/installedApps/cell_name/OracleRTD.ear`.
- If you are using WebLogic, `RTD_RUNTIME_HOME` is the directory where you put the contents of the `RTD.ear` file. For example, if you are using WebLogic and your `RTD_HOME` is `C:\OracleBI\RTD`, your `RTD_RUNTIME_HOME` is `C:\OracleBI\RTD\package\expanded`.

## 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.

The following sections provide information about creating a database, according to database type, as well as information about initializing the Oracle RTD Database.

### 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 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 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 SDDDBTool

You initialize the Oracle RTD Database using SDDDBTool, an Oracle RTD Windows client application. Before using SDDDBTool to initialize the database, make sure that Real-Time Decision Server is not running.

To initialize the Oracle RTD Database using SDDDBTool:

- 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 SDDDBTool by running `RTD_HOME\scripts\SDDDBTool.cmd` on the computer where you installed the Oracle RTD client tools. Then, choose your database type.
- 4 Enter your database settings, as follows:
  - Database Host
  - Database Port
  - Database Name (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 datasource 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 the following section 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 two tables, `CrossSellCustomers` and `CrossSellResponses`, 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`.

- If you are using Oracle Database for your Oracle RTD Database, run `RTD_HOME\examples\CrossSell\etc\data\Oracle\InitAppDB`.
- If you are using DB2 for your Oracle RTD Database, run `RTD_HOME\examples\CrossSell\etc\data\DB2\InitAppDB`.

This script takes the following parameters:

```
InitAppDB RTD_HOME db_host db_port db_name db_runtime_user
db_admin_user db_admin_password
```

Where:

`RTD_HOME` is the full path of the directory where the Oracle RTD files are installed

`db_host` is the name of the computer hosting the database server

`db_port` is the database port number

`db_name` is the name of the database, or for Oracle Database, the SID

`db_runtime_user` is the user name of the run-time user for the system

`db_admin_user` is the name of a user that has rights to create tables and stored procedures on the database

`db_admin_password` is the password of the administrative user

**Note:** 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 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 default 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`.

- 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
```

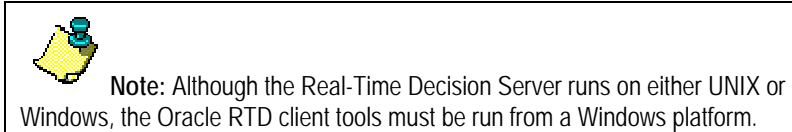
Make sure to replace *RTD\_HOME* with the actual Oracle RTD installation path. For *new\_password*, provide the password you want to use for the server private key.

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

## 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.



This section contains the following topics:

- 3.1 Configuring server properties
- 3.2 Creating a JDBC provider for the Oracle RTD Database
- 3.3 Installing the Oracle Real-Time Decisions application on OC4J
- 3.4 Starting Oracle Real-Time Decisions
- 3.5 Uninstalling the Oracle Real-Time Decisions application from OC4J
- 3.6 Configuring SSL for Real-Time Decision Server (recommended)
- 3.7 Setting up JConsole for OC4J
- 3.8 Changing the Oracle Real-Time Decisions port number

### 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.

#### 3.1.1 Configuring server properties for standalone OC4J

- 1 Follow the steps appropriate to your operating system:
  - a. **On Windows:** Locate the file `OC4J_HOME\bin\oc4j.cmd` and open it for editing. After the line `:oc4j`, insert the following four 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
```

Then, save and close the file.

**Tip:** To reduce the likelihood of getting out-of-memory errors, you may want to increase the JVM memory at this point. To do this, add an additional line to the `oc4j.cmd` file, as follows:

```
set JVMARGS=%OC4J_JVM_ARGS% -Xms512M -Xmx1024M
```

- b. **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 four 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
```

If you installed OC4J on an AIX-based system, add the following additional line:

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

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.

**Tip:** To reduce the likelihood of getting out-of-memory errors, you may want to increase the JVM memory at this point. To do this, add an additional line to the `oc4j` file, as follows:

```
OC4J_JVM_ARGS=$OC4J_JVM_ARGS -Xms512M -Xmx1024M
```

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 `display-name` attribute to `OC4J 10g (10.1.3) 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/schem
a/web-site-10_0.xsd" port="8080" 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" />
<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:
  - a. **For SQL Server:** Copy the file `sqljdbc.jar` in the directory `RTD_HOME/lib/jdbc/` to the directory `OC4J_HOME/j2ee/home/applib`
  - b. **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

- 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
```

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" />
      </category>
    </module-data>
  </process-type>
</ias-component>
```

**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.

If you installed Oracle Application Server on an AIX-based system, include the following additional Java option:

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

**Note:** Make sure to include a space between this Java option and the preceding options.

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 10g (10.1.3) 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/schem
a/web-site-10_0.xsd" port="8080" 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" />
<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:
  - a. **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`
  - b. **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 Creating a JDBC provider for the Oracle RTD Database

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

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

- 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, 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.
- 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.

**Note:** 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:
  - For **SQL Server**: `com.microsoft.sqlserver.jdbc.SQLServerDriver`
  - For **Oracle Database**: `oracle.jdbc.driver.OracleDriver`
  - For **DB2**: `com.ibm.db2.jcc.DB2Driver`

**Note:** If you are using Oracle Database for your Oracle RTD Database, you *must* enter `oracle.jdbc.driver.OracleDriver` for **Connection Factory Class**. Do not use the default value of `oracle.jdbc.pool.OracleDataSource`.

- 7 For **JDBC URL**, enter one of the following:
  - For **SQL Server**:  
`jdbc:sqlserver://db_host:db_port;databaseName=db_name`
  - For **Oracle Database**: `jdbc:oracle:thin:@db_host:db_port:sid`
  - For **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 `rttd`), 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 Containers for J2EE 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:
  - a. If you are using Oracle Database for your Oracle RTD Database, keep the default settings and click **Test**.
  - b. 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.

### 3.3 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.

- 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 On the OC4J home page, click the **Applications** tab.

**Note:** 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:8080/ui`.

### 3.4 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.

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

## 3.5 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.

- 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**. Then, on the Confirmation page, click **Yes**. A red arrow pointing down in the Status column indicates that the application is stopped.  
**Note:** 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.

## 3.6 Configuring SSL for Real-Time Decision Server (recommended)

Follow the steps in this section to set up SSL for all client connections to the Real-Time Decision Server. Before you begin, ensure that you followed the instructions in Section 2.4 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 2.4.

- 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.
  - a. **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%
```
  - b. **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 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:
  - a. In the `<web-site>` tag, set the `port` attribute to `8443`.
  - b. In the `<web-site>` tag, add or set the `secure` attribute to `true`.

- c. In the `<web-site>` tag, set the `display-name` attribute to OC4J 10g (10.1.3) RTD Secure Web Site.
- d. For each `<web-app>` tag, add or set the `shared` attribute to `true`.
- e. In the `<access-log>` tag, set the `path` attribute to `../log/rtd-secure-web-access.log`.
- f. 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/schem
a/web-site-10_0.xsd" port="8443" secure="true" display-name="OC4J 10g
(10.1.3) 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="RTD_HOME/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" />
```

- 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, follow these steps:

1. Use JConsole to change the **WebServerPort** attribute of the OracleRTD > SDPropertyManager > Misc MBean from the HTTP port to the SSL port. Then, set the **HTTPSEnabled** attribute to `true`. See Section 10.1, Accessing JConsole for more information.

2. 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.

### 3.6.1 Testing the SSL configuration

To verify that the SSL port is functioning properly, go to Decision Center at the URL `http://server_name:ssl_port/ui`. If the SSL port is functioning properly, 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.

## 3.7 Setting up JConsole for OC4J

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

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

- a. **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.

- b. **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.

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

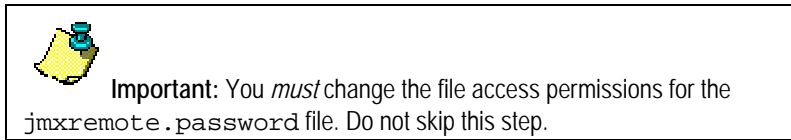
```
monitorRole readonly
controlRole readwrite
```

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:

- a. **On Windows:** Follow the instructions at <http://java.sun.com/j2se/1.5.0/docs/guide/management/security-windows.html>
- b. **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>.



- 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 10.1, Accessing JConsole for more information.

## 3.8 Changing the Oracle Real-Time Decisions port number

To change the Oracle RTD application port number, you must change the port number in both OC4J and in Oracle RTD.

### 3.8.1 Changing the port number in OC4J

- 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.

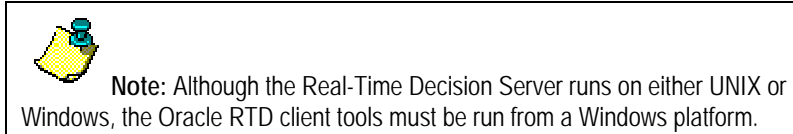


### 3.8.2 Changing the port number in Oracle RTD

- 1 Open JConsole by running `jconsole` from your JDK installation directory. See 10.1, Accessing JConsole for more information.
- 2 Click the **Remote** tab, then enter the following:
  - a. For **Host**, enter the name of the computer where Real-Time Decision Server is running.
  - b. For **Port**, enter your JMX remote port number (for example, 12345).
  - c. For **User Name** and **Password**, enter the credentials you specified when you set up JConsole. For more information, see Section 3.7, Setting up JConsole for OC4J.
  - d. Click **Connect**.
- 3 Click the **MBeans** tab and go to the **Oracle RTD > SDPropertyManager > Misc MBean**.
- 4 Change the **WebServerPort** to the new port number, then press **Enter**.
- 5 Close JConsole.
- 6 Restart Oracle RTD.

## 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.



This section contains the following topics:

- 4.1 Configuring server properties
- 4.2 Creating a JDBC provider for the Oracle RTD Database
- 4.3 Installing the Oracle Real-Time Decisions application on WebSphere
- 4.4 Starting Oracle Real-Time Decisions
- 4.5 Uninstalling the Oracle Real-Time Decisions application from WebSphere
- 4.6 Configuring SSL for Real-Time Decision Server (recommended)
- 4.7 Setting up JConsole for WebSphere
- 4.8 Changing the Oracle Real-Time Decisions port number

### 4.1 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.

- 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 If you installed WebSphere on an AIX-based system, follow these steps to add a parameter to the generic JVM arguments:
  - a. 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.

- b. 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).
- 7 On the Process Definition page, click **Java Virtual Machine**, then click **Custom Properties** under the Additional Properties heading.
- 8 Click **New**.
- 9 For **Name**, enter `org.eclipse.emf.ecore.EPackage.Registry.INSTANCE`.
- 10 For **Value**, enter `com.sigmadynamics.emf.util.SDEMFRegistry`.
- 11 Click **OK**, then click **Save**.
- 12 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 13 and 14 and proceed to Step 15. If you want to add a new HTTP transport port for Oracle RTD, continue to Step 13.

- 13 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**.
- 14 If you added a new HTTP transport port in Step 13, follow these steps to create a new host alias. Otherwise, continue to Step 15.
- 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**.
- 15 If you are not using 8080 as the Oracle RTD application port, you must update the port number in Oracle RTD. See Section 4.8.2, Changing the port number in Oracle RTD for more information. Note that you cannot perform the steps described in Section 4.8.2 until you complete the tasks in Sections 4.2, 4.3, 4.4, and 4.7.
- 16 Restart WebSphere.

## 4.2 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.

- 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_nameNode_number`), then click **New**.
- 4 For **Database type**, select `User-defined`.
- 5 For **Implementation class name**, enter one of the following values:
  - a. For **SQL Server**: `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`
  - b. For **Oracle Database**: `oracle.jdbc.pool.OracleConnectionPoolDataSource`
  - c. 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:
  - a. For **SQL Server**: `RTD_HOME/lib/jdbc/sqljdbc.jar`

- b. For Oracle Database: `RTD_HOME/lib/jdbc/ojdbc14.jar`
- c. For DB2: `RTD_HOME/lib/jdbc/db2jcc.jar`; `RTD_HOME/lib/jdbc/db2jcc_license_cu.jar`

Make sure to replace `RTD_HOME` with the actual Oracle RTD installation path.

- 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 Summary page, click **Finish**, then click **Save**.
- 14 Click the name of the data source you created (for example, `RTD_DS`).
- 15 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**. Then, click **OK**.
  - f. Click **Save**.
- 16 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**.
- 17 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**.
- 18 Restart WebSphere, then launch the Integrated Solutions Console again.
- 19 Expand **Resources**, then expand **JDBC** and choose **Data Sources**.  
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.

### 4.3 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.

- 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 Under the Map resource references to resources heading, select **Use default method (many-to-one mapping)**.

- 8 Select the J2C authentication you created in the previous section (RTDDS\_auth). Click the **Select All** icon, then click **Apply**.
- 9 For each module, set the **Target Resource JNDI Name** to SDDS.
- 10 Click **Next**, then click **Next** again.
- 11 Click **Finish**, then click **Save**.

## 4.4 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.

- 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:8080/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.

## 4.5 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.

- 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.

## 4.6 Configuring SSL for Real-Time Decision Server (recommended)

Follow the steps in this section to set up SSL for all client connections to the Real-Time Decision Server. Before you begin, ensure that you followed the instructions in Section 2.4 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 2.4.

- 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 2.4.
  - 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 2.4.
  - 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, follow these steps:

1. Use JConsole to change the **WebServerPort** attribute of the OracleRTD > SDPropertyManager > Misc MBean, from the HTTP port to the SSL port. Then, set the **HTTPSEnabled** attribute to `true`. See Section 10.1, Accessing JConsole for more information.
2. Disable the HTTP port for your Web server using application server tools. Refer to the WebSphere documentation for more information.

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

#### 4.6.1 Testing the SSL configuration

To verify that the SSL port is functioning properly, go to Decision Center at the URL `http://server_name:ssl_port/ui`. If the SSL port is functioning properly, 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.

## 4.7 Setting up JConsole for WebSphere

Perform the steps in the following sections to set up the JConsole management tool.

### 4.7.1 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.

#### 4.7.2 Setting up a batch file for JConsole

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

```
set WAS_HOME=WEBSPPHERE_HOME\AppServer
set USER_HOME=WEBSPPHERE_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 example, 12345). For WebSphere, the JMX remote port is always the same as the WebSphere bootstrap port. Make sure to replace `WEBSPPHERE_HOME` with the actual WebSphere installation path, and replace `profile_name` with the name of your WebSphere profile (for example, `AppSrv01`).

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

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

```
#!/bin/sh
WAS_HOME=WEBSPPHERE_HOME/AppServer
USER_HOME=WEBSPPHERE_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:$WA
S_HOME/java/lib/tools.jar -J-
Dcom.ibm.CORBA.ConfigURL=file:$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 `WEBSPPHERE_HOME` with the actual WebSphere installation path, and replace `profile_name` with the name of your WebSphere profile (for example, `AppSrv01`).

**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.

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

```
#!/bin/sh
WAS_HOME=WEBSHERE_HOME/AppServer
USER_HOME=WEBSHERE_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 `WEBSHERE_HOME` with the actual WebSphere installation path, and replace `profile_name` with the name of your WebSphere profile (for example, `AppSrv01`).

**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.

#### 4.7.3 Creating 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**, then provide the following values:
  - a. For **User**, enter the user ID you provided in Step 3.
  - b. For **Role(s)**, select **Administrator**.
- 7 Click **OK**, then click **Save**.

You can now run JConsole and log in, using the user ID and password you just created.. See Section 10.1, Accessing JConsole for more information.

## 4.8 Changing the Oracle Real-Time Decisions port number

To change the Oracle RTD application port number, you must change the port number in both WebSphere and in Oracle RTD.

#### 4.8.1 Changing the port number in WebSphere

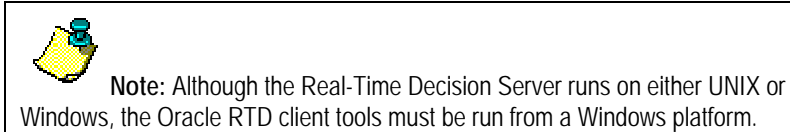
- 1 Access the Integrated Solutions Console at the URL `http://websphere_host:port/ibm/console`. At the login prompt, enter the administrator user name and password. On Windows, you can also access the Integrated Solutions Console through **Start > Programs**.
- 2 In the tree on the left, expand **Servers** 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.

#### 4.8.2 Changing the port number in Oracle RTD

- 1 On the computer where WebSphere is installed, run the batch script you created in Section 4.7.2, Setting up a batch file for JConsole.
- 2 Log in as the JConsole user you created in Section 4.7.3, Creating a JConsole user (for example, `jmx_admin`).
- 3 Click the **MBeans** tab and go to the **Oracle RTD > SDPropertyManager > Misc MBean**.
- 4 Change the **WebServerPort** to the new Oracle RTD application port number, then press **Enter**.
- 5 Close JConsole.
- 6 Restart Oracle RTD.

## 5 Configuring WebLogic for Oracle Real-Time Decisions

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



This section contains the following topics:

- 5.1 Configuring server properties
- 5.2 Creating a JDBC provider for the Oracle RTD Database
- 5.3 Installing the Oracle Real-Time Decisions application on WebLogic
- 5.4 Starting Oracle Real-Time Decisions
- 5.5 Uninstalling the Oracle Real-Time Decisions application from WebLogic
- 5.6 Configuring SSL for Real-Time Decision Server (recommended)
- 5.7 Setting up JConsole for WebLogic
- 5.8 Changing the Oracle Real-Time Decisions port number

### 5.1 Configuring server properties

Use the WebLogic administrative console, called the WebLogic Server Administration Console, to configure server properties. For more information about how to use the WebLogic Server Administration Console, refer to the WebLogic documentation.

- 1 Start WebLogic, as follows:
  - **On Windows:** Use the WebLogic startup script at `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.cmd`
  - **On UNIX:** Use the WebLogic startup script at `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh`



**Note:** If you are running WebLogic on an AIX-based system, you must perform the following steps before starting WebLogic:

1. Open the `BEA_HOME/Weblogic92/common/bin/commEnv.sh` file for editing.
2. Add the following additional line:  
`JAVA_OPTIONS=" $JAVA_OPTIONS -Dcom.sun.xml.name space.QName.useCompatibleSerialVersionUID=1.0 "`
3. Save and close the file.

For more information, see the Known Issues for WebLogic on AIX-based systems at: [http://e-docs.bea.com/platform/suppconfigs/configs/ibm\\_aix/ibm\\_aix53\\_92.html#97122](http://e-docs.bea.com/platform/suppconfigs/configs/ibm_aix/ibm_aix53_92.html#97122)

2. 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 > BEA Products > User Projects > domain\_name > Admin Server Console**.
3. Create a new channel for the Oracle RTD application port, as follows:
  - a. In the tree on the left, expand **Environment** and choose **Servers**.
  - b. Click the name of the server where you plan to deploy Oracle RTD (for example, **AdminServer**).
  - c. Click the **Protocols** tab, then click the **Channels** subtab.
  - d. Click **New**. You may need to click **Lock & Edit** first to enable the **New** button.
  - e. For **Name**, enter `OracleRTD_port` or some similar value. Then, select **t3** for **Protocol** (the default) and click **Next**.
  - f. For **Listen Port** and **External Listen Port**, enter the port number you want to use for Oracle RTD. By default, the Oracle RTD application runs on port 8080.
  - g. Click **Finish**.
  - h. If you are not using 8080 as the Oracle RTD application port, you must update the port number in Oracle RTD. See Section 5.8.2, Changing the port number in Oracle RTD, for more information. Note that you cannot perform the steps described in Section 5.8.2 until you complete the tasks in Sections 5.2, 5.3, and 5.7.
4. Stop WebLogic:
  - On **Windows**, open a new command window and run the following command:  
`BEA_HOME\user_projects\domains\domain_name\bin\stopWebLogic.cmd`
  - On **UNIX**, open a new terminal window and run the following command:  
`BEA_HOME/user_projects/domains/domain_name/bin/stopWebLogic.sh`
5. If you installed WebLogic on Windows, follow these steps:
  - a. Make a backup copy of the file `BEA_HOME\user_projects\domains\domain_name\bin\startWebLogic.cmd`.

- b. Open `startWebLogic.cmd` for editing. At the top of the file, insert the following new lines:

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dorg.eclipse.emf.ecore.EPackage.Registry.INSTANCE=com.sigmadynamics.emf.util.SDEMFRegistry
set JAVA_OPTIONS=%JAVA_OPTIONS% -Djavax.xml.parsers.SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXParserFactoryImpl
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote=true
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.port=your_jmx_remote_port
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.authenticate=false
set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.sun.management.jmxremote.ssl=false
```

For `jmx_remote_port`, enter the JConsole port number (for example, 12345).

- c. In addition to the `JAVA_OPTIONS` described in the previous section, you must also set the classpath to include the path to the JDBC jar files. The jar file names vary, depending on the type of database you are using for your Oracle RTD Database.

For SQL Server, include the following additional line:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\sqljdbc.jar;
```

For Oracle Database, include the following additional line:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\ojdbc14.jar;
```

For DB2, include the following two additional lines:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\db2jcc.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\db2jcc_license_cu.jar;
```

Make sure to replace `RTD_HOME` with the actual Oracle RTD installation path.

- d. Save and close the file.
- 6 If you installed WebLogic on UNIX, follow these steps:
- Make a backup copy of the file `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh`.
  - Open `startWebLogic.sh` for editing. At the top of the file, insert the following new lines:



```

JAVA_OPTIONS="{JAVA_OPTIONS} -Dorg.eclipse.emf.ecore.EPackage.
Registry.INSTANCE=com.sigmadynamics.emf.util.SDEMFRegistry"
JAVA_OPTIONS="{JAVA_OPTIONS} -Djava.awt.headless=true"
JAVA_OPTIONS="{JAVA_OPTIONS} -Djavax.xml.parsers.
SAXParserFactory=com.sun.org.apache.xerces.internal.jaxp.SAXParser
FactoryImpl"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote=true"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.
port=your_jmx_remote_port"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.
authenticate=false"
JAVA_OPTIONS="{JAVA_OPTIONS} -Dcom.sun.management.jmxremote.
ssl=false"
SAVE_JAVA_OPTIONS=" "

```

For *jmx\_remote\_port*, enter the JConsole port number (for example, 12345).

If you installed WebLogic on an AIX-based system, add the following additional line before `SAVE_JAVA_OPTIONS=" "`:

```

JAVA_OPTIONS="{JAVA_OPTIONS} -Djava.net.preferIPv4Stack=true"

```

- c. In addition to the `JAVA_OPTIONS` described in the previous section, you must also set the classpath to include the path to the JDBC jar files. The jar file names vary, depending on the type of database you are using for your Oracle RTD Database.

For SQL Server, include the following additional line:

```

EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/sqljdbc.
jar"

```

For Oracle Database, include the following additional line:

```

EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/ojdbc14.
jar"

```

For DB2, include the following two additional lines:

```

EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc.
jar"
EXT_PRE_CLASSPATH="{EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc_
license_cu.jar"

```

Make sure to replace *RTD\_HOME* with the actual Oracle RTD installation path.

- d. Save and close the file.
- 7 Start WebLogic using the revised startup scripts, as follows:
- **On Windows:** `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.cmd`
  - **On UNIX:** `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh`

## 5.2 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.

- 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 > BEA Products > User Projects > domain\_name > Admin Server Console**.
- 2 In the tree on the left, expand **Services**, then expand **JDBC** and choose **Data Sources**.
- 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**, enter `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**.
- 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 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**.
- 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.driver.OracleDriver`
  - **DB2:** `com.ibm.db2.jcc.DB2Driver`
- 8 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`
  - **Oracle Database:** `jdbc:oracle:thin:@db_host:db_port:sid`
  - **DB2:** `jdbc:db2://db_host:db_port/db_name`
- 9 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`
- 10 Scroll to the bottom of the page and enter `SDAPPS` for **Test Table**.

- 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 data source to be made available (for example, **AdminServer**). You must perform this step before completing the data source configuration.
- 13 Click **Finish**.
- 14 Click **Activate Changes**.

### 5.3 Installing the Oracle Real-Time Decisions application on WebLogic

Use the WebLogic Server Administration Console to install Oracle RTD on WebLogic. Before you begin, ensure that WebLogic is started.

- 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 > BEA Products > User Projects > domain\_name > Admin Server Console**.
- 2 In the tree on the left, click **Deployments**.
- 3 Click **Install**. You may need to click **Lock & Edit** first to enable the **Install** button.
- 4 Select the file path where you expanded the RTD .ear file (for example, `RTD_HOME/package/expanded`), then click **Next**. You expanded the RTD.ear file in Section 2.1.
- 5 Select **Install this deployment as an application**, then click **Next**.
- 6 On the Optional Settings page, enter `OracleRTD` for **Name**. Then, click **Next**.
- 7 Review your settings and click **Finish**.
- 8 Click **Save**, then click **Activate Changes**.
- 9 Restart WebLogic.
- 10 Access the WebLogic Server Administration Console again.
- 11 In the tree on the left, click **Deployments**.
- 12 Verify in the list of deployments that the **State** column for the application `OracleRTD` is **Active**. If the status is **Prepared**, then you must start Oracle RTD. If you encounter problems, check the WebLogic logs for errors, then resolve any issues and restart WebLogic.
- 13 Check to see if Oracle RTD is running by going to Decision Center at the URL `http://server_name:8080/ui`. The Decision Center login page appears.

Note that authentication has not yet been enabled for Decision Center. To enable authentication, refer to Section 6, Configuring Authentication for Oracle Real-Time Decisions.

### 5.4 Starting Oracle Real-Time Decisions

After you install Oracle RTD on WebLogic, Oracle RTD is not started by default. You can use the Weblogic Server Administration Console to start Oracle RTD. Before you begin, ensure that WebLogic is started.

- 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 > BEA Products > User Projects > *domain\_name* > Admin Server Console**.
- 2 In the tree on the left, click **Deployments**.
- 3 Select **OracleRTD** and click **Start**. You may need to click **Lock & Edit** first.

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

## 5.5 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.

- 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 > BEA Products > 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**. When prompted to confirm the action, click **Yes**.
- 4 Select **OracleRTD** and click **Delete**. When prompted to confirm the action, click **Yes**.
- 5 Click **Activate Changes**.

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


## 5.6 Configuring SSL for Real-Time Decision Server (recommended)

Follow the steps in this section to set up SSL for all client connections to the Real-Time Decision Server. Before you begin, ensure that you followed the instructions in Section 2.4 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 2.4.

- 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 > BEA Products > 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 Administration Server (typically **AdminServer**). You may need to click **Lock & Edit** first.
- 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 2.4.
  - 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 2.4.
- 6 Click the **SSL** tab, then provide the following values:
  - a. For **Identity and Trust Locations**, select **Keystores**.
  - b. For **Private Key Alias**, **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 2.4.
- 7 Click **Save**, then click **Activate Changes**.
- 8 Restart WebLogic.



**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, follow these steps:

1. Use JConsole to change the **WebServerPort** attribute of the OracleRTD > SDPropertyManager > Misc MBean, from the HTTP port to the SSL port. Then, set the **HTTPSEnabled** attribute to `true`. See Section 10.1, Accessing JConsole for more information.
2. Disable the HTTP port for your Web server using application server tools. Refer to the WebLogic documentation for more information.

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

### 5.6.1 Testing the SSL configuration

To verify that the SSL port is functioning properly, go to Decision Center at the URL `http://server_name:ssl_port/ui`. If the SSL port is functioning properly, 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.

## 5.7 Setting up JConsole for WebLogic

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

- 1 Follow the steps appropriate for your operating system:

- a. If you installed WebLogic on Windows, open `BEA_HOME\user_projects\domains\domain_name\bin\startWebLogic.cmd` for editing. Locate the line:

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

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

- b. If you installed WebLogic on UNIX, open `startWebLogic.sh` for editing. Locate the line:

```
JAVA_OPTIONS="$ {JAVA_OPTIONS} -Dcom.sun.management.jmxremote.  
authenticate=false"
```

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

- 2 Go to `BEA_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.

- 3 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. The user name `monitorRole` allows JConsole MBean read-only permissions, while the user name `controlRole` allows for full JMX MBean control. Change the passwords to new ones of your choice. Optionally, you can add new user names and passwords on separate lines.

- 4 You can also set permissions for each set of credentials. To do this, 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:

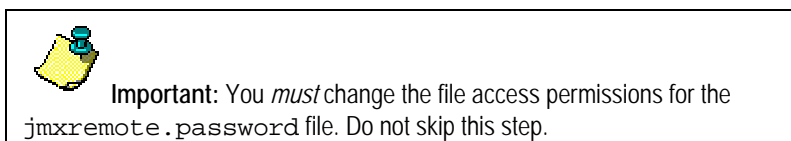
```
monitorRole readonly
controlRole readwrite
```

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

- 5 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:

- a. **On Windows:** Follow the instructions at <http://java.sun.com/j2se/1.5.0/docs/guide/management/security-windows.html>
- b. **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>.



- 6 Start WebLogic.

You can now run JConsole and log in, using the user name and password combination defined in the `jmxremote.password` file. See Section 10.1, Accessing JConsole, for more information.

## 5.8 Changing the Oracle Real-Time Decisions port number

To change the Oracle RTD application port number, you must change the port number in both WebLogic and in Oracle RTD.

### 5.8.1 Changing the port number 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 > BEA Products > 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 server where Oracle RTD is running (for example, **AdminServer**).
- 4 Click the **Protocols** tab, then click the **Channels** subtab.
- 5 Click the name of the Oracle RTD channel (for example, **OracleRTD\_port**).
- 6 Deselect **Enabled**. You may need to click **Lock & Edit** first.
- 7 For **Listen Port**, enter the new port number you want to use for Oracle RTD.
- 8 Select **Enabled**.
- 9 Click **Save**, then click **Activate Changes**.
- 10 Restart WebLogic.

### 5.8.2 Changing the port number in Oracle RTD

- 1 Open JConsole by running `jconsole` from your JDK installation directory. See 10.1, Accessing JConsole for more information.
- 2 Click the **Remote** tab, then enter the following:
  - a. For **Host**, enter the name of the computer where Real-Time Decision Server is running.
  - b. For **Port**, enter your JMX remote port number (for example, 12345).
  - c. For **User Name** and **Password**, enter the credentials you specified when you set up JConsole. For more information, see Section 5.7, Setting up JConsole for WebLogic.
  - d. Click **Connect**.
- 3 Click the **MBeans** tab and go to the **Oracle RTD > SDPropertyManager > Misc MBean**.
- 4 Change the **WebServerPort** to the new port number, then press **Enter**.
- 5 Close JConsole.
- 6 Restart Oracle RTD.



## 6 Configuring Authentication for Oracle Real-Time Decisions

Oracle RTD has authentication on both the server (cluster) and Inline Service levels. Three types of authentication are available for Oracle RTD: Windows Authentication, Oracle RTD Platform Authentication, and Oracle's Siebel Object Manager Authentication. Authentication is *not* enabled by default. Follow the instructions in this section to enable and configure authentication for Oracle RTD.

This section contains the following topics:

- 6.1 Oracle Real-Time Decisions security overview
- 6.2 Configuring Windows Authentication
- 6.3 Configuring Oracle RTD Platform Authentication
- 6.4 Configuring Oracle's Siebel Object Manager Authentication
- 6.5 Assigning permissions to users and groups
- 6.6 Changing the authentication type

### 6.1 Oracle Real-Time Decisions security overview

Oracle RTD has three different levels of security: server-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.

Server-wide security is configured using the SecurityManager MBean. You can use JConsole to set permission codes for Oracle RTD users. See Section 10.5, Oracle Real-Time Decisions security management for more information.

Inline Service security is configured on individual Inline Services. Use Decision Studio to set Inline Service security. See *Decision Studio Reference Guide* for more information.

Perspective-level security is also configured in Decision Studio. See *Decision Studio Reference Guide* for more information.

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

This section contains the following additional security topics:

- 6.1.1 Oracle RTD default accounts and passwords
- 6.1.2 Oracle RTD configuration and log files
- 6.1.3 Oracle RTD cookies

#### 6.1.1 Oracle RTD default accounts and passwords

By default, Oracle RTD has the following default accounts and passwords:

- If you are using OC4J or WebLogic, you set up authentication for JConsole using the `jmxremote.password` file, as described in Sections 3.7 and 5.7. This file contains two default accounts, `monitorRole` and `controlRole`. The default passwords for these accounts are `QED` and `R&D`, respectively.
- If you are using SSL for client connections to Oracle RTD, the default password for the keystore, truststore, and server private key is `tc-ssl`. If you want to use the default Oracle RTD keystore and truststore, you must change these passwords before using SSL with Oracle RTD; see Section 2.4 for more information.

### 6.1.2 Oracle RTD configuration and log files

The following sections provide the names and locations of the Oracle RTD configuration and log 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.

#### 6.1.2.1 Oracle RTD configuration and log files for standalone OC4J

Configuration files:

`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`

Log files:

**`OC4J_HOME/j2ee/home/application/OracleRTD/log/server.log`**

`OC4J_HOME/j2ee/home/application-deployments/OracleRTD/application.log`

`OC4J_HOME/j2ee/home/log/server.log`

`OC4J_HOME/j2ee/home/log/oc4j/log.xml`

The Oracle RTD `server.log` shown in bold is the main Oracle RTD log, and is the primary log to use for troubleshooting problems.

#### 6.1.2.2 Oracle RTD configuration and log files for Oracle Application Server

Configuration files:

`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-tmpl.xml`

`ORACLE_AS_HOME/j2ee/oc4j_instance/applications/OracleRTD/etc/sdlog4j.xml`

Log files:

*ORACLE\_AS\_HOME/j2ee/oc4j\_instance/application-deployments/OracleRTD/home\_default\_group\_1/application.log*

***ORACLE\_AS\_HOME/j2ee/oc4j\_instance/applications/OracleRTD/log/server.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*

The Oracle RTD *server.log* shown in bold is the main Oracle RTD log, and is the primary log to use for troubleshooting problems.

### 6.1.2.3 Oracle RTD configuration and log files for WebSphere

Configuration files:

*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-tmpl.xml*

*WEBSPHERE\_HOME/AppServer/profiles/profile\_name/installedApps/host\_node\_cell\_name/OracleRTD.ear/etc/sdlog4j.xml*

Log files:

*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)

***WEBSPHERE\_HOME/AppServer/profiles/profile\_name/installedApps/host\_node\_cell\_name/OracleRTD.ear/log/server.log***

The Oracle RTD *server.log* shown in bold is the main Oracle RTD log, and is the primary log to use for troubleshooting problems.

#### 6.1.2.4 Oracle RTD configuration and log files for WebLogic

Configuration files:

*RTD\_HOME/package/expanded/etc/sdconfig-cluster.xml*

*RTD\_HOME/package/expanded/etc/sdconfig-machine.xml*

*RTD\_HOME/package/expanded/etc/sdconfig-tmpl.xml*

*RTD\_HOME/package/expanded/etc/sdlog4j.xml*

Log files:

*BEA\_HOME/user\_projects/domains/domain\_name/servers/server\_name/logs/server\_name.log*

*BEA\_HOME/user\_projects/domains/domain\_name/servers/server\_name/logs/profile\_name.log*

***RTD\_HOME/package/expanded/log/server.log***

The Oracle RTD *server.log* shown in bold is the main Oracle RTD log, and is the primary log to use for troubleshooting problems.

#### 6.1.2.5 Configuration and log files for Oracle RTD client tools

Configuration files:

*RTD\_HOME/client/clientHttpEndpoints.properties*

*RTD\_HOME/client/clientHttpsEndpoints.properties*

Log files:

*RTD\_HOME/scripts/SDDBTool.log*

*RTD\_HOME/log/client.log*

*RTD\_HOME/log/loadgen.csv*

*RTD\_HOME/log/server.log*

### 6.1.3 Oracle RTD cookies

Oracle RTD generates and stores one non-persistent cookie on client computers for the duration of a login session. This cookie does not contain personally identifiable information; instead, it contains session information that lets Oracle RTD avoid requiring the user to log in for every request. Full details are as follows:

**Cookie name:** sdLoginTicket

**Path:** /ui

**Description:** Contains a base 64-encoded string representation of the Oracle RTD session table index, as well as an SHA-1 hash of the user credentials and four random bytes. Real-Time Decision Server validates the ticket by using its

session table index to locate the session, verifying that the session has not expired, and then verifying that the session contains the same hash.

## 6.2 Configuring Windows Authentication

On the computer running Real-Time Decision Server, follow the steps in this section to configure Windows Authentication for Oracle RTD. Real-Time Decision Server must be running on a Windows computer in order to use Windows Authentication for Oracle RTD.

- 1 Download the Tagishauth utility from <http://free.tagish.net/jaas/>. Make sure to download version 1.0.3.
- 2 Extract `tagishauth.jar` and put it in `RTD_HOME\lib`.
- 3 Extract `NTSystem.dll` into the same directory.
- 4 Edit the Windows `PATH` environment variable to include the location of `NTSystem.dll` (`RTD_HOME\lib`). Make sure to replace `RTD_HOME` with the actual Oracle RTD installation path. The location of `NTSystem.dll` must be added to the *beginning* of the system `PATH`. You should only provide the folder location of `NTSystem.dll`; do not include the actual filename. For example:  

```
C:\OracleBI\RTD\lib
```
- 5 If you are running Real-Time Decision Server on OC4J, copy the file `tagishauth.jar` from `RTD_HOME\lib` to `OC4J_HOME\j2ee\home\applib`. Then, restart OC4J.
- 6 If you are running Real-Time Decision Server on WebSphere, follow these steps:
  - a. 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**.
  - b. In the tree on the left, expand **Environment** and choose **Shared Libraries**.
  - c. Click **New**. Provide a name for the Shared Library (such as `Tagish Lib`), then enter `RTD_HOME\lib\tagishauth.jar` for **Classpath**, making sure to replace `RTD_HOME` with the actual Oracle RTD installation path. Then, click **OK**.
  - d. In the tree on the left, expand **Applications**, then expand **Enterprise Applications** and choose **OracleRTD**.
  - e. Scroll to the bottom of the screen and expand **Additional Properties**, then click **Libraries**.
  - f. Click **Add**. Select the name of your Shared Library from the drop-down list, then click **OK**.
  - g. Click **Save**.
- 7 If you are running Real-Time Decision Server on WebLogic, follow these steps:
  - a. Go to the `BEA_HOME\user_projects\domains\domain_name\bin` and open the `startWebLogic_RTD.cmd` file for editing.

- b. Add the following new line to the top of the file:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;%RTD_HOME%\lib\tagishauth.jar
```

Make sure to replace *RTD\_HOME* with the actual Oracle RTD installation path.

- c. Save and close the file.
- 8 Start Oracle RTD.
- 9 Start JConsole and log in. See Section 10.1, Accessing JConsole, for more information.
- 10 Go to the **OracleRTD > SDClusterPropertyManager > Windows Authenticator Properties** MBean. Then, set the **WindowsAuthenticatorDomain** property to your domain or computer name.
- 11 Go to the **OracleRTD > SDClusterPropertyManager > Security Properties** MBean. Then, set the **AuthenticationProviderClass** property to `com.sigmadynamics.server.security.WindowsAuthenticator` and set the **AuthenticationEnabled** property to `true`.

When you are using Windows Authentication, use the Windows operating system to create users and groups for Oracle RTD. Then, assign Oracle RTD permissions to those users and groups using JConsole. See Section 6.5, Assigning permissions to users and groups, for more information.

### 6.3 Configuring Oracle RTD Platform Authentication

Follow the steps in this section to configure Oracle RTD Platform Authentication for Oracle RTD, and to manage users and groups for this type of authentication.

To set up Oracle RTD Platform Authentication:

- 1 Start Oracle RTD.
- 2 Start JConsole and log in. See Section 10.1, Accessing JConsole, for more information.
- 3 Go to the **OracleRTD > SDClusterPropertyManager > Security Properties** MBean.
- 4 Set the **AuthenticationEnabled** property to `true`.
- 5 Set the **AuthenticationProviderClass** property to `com.sigmadynamics.server.security.DBAuthenticator`.

To manage users and groups for Oracle RTD Platform Authentication:

- 1 Start Oracle RTD.
- 2 Start JConsole and log in. See Section 10.1, Accessing JConsole, for more information.
- 3 Go to the **OracleRTD > SDManagement > SecurityManager** MBean.
- 4 To create new Oracle RTD users, use the `createUser()` operation. Only use this method to create users if you are using Oracle RTD Platform Authentication.
- 5 To create new security groups, use the `createGroup()` operation. To add users to Oracle RTD groups, use the `addUserToGroup()` operation.

## 6.4 Configuring Oracle's Siebel Object Manager Authentication

The following sections provide information about how to configure Siebel Object Manager Authentication for Oracle RTD:

- 6.4.1 Providing access to jar files for your application server
- 6.4.2 Enabling authentication in Oracle RTD

### 6.4.1 Providing access to jar files for your application server

Perform the steps in the following sections to provide access to Siebel jar files for your application server. After you complete the steps to provide access to jar files, proceed to Section 6.4.2, Enabling authentication in Oracle RTD.

#### 6.4.1.1 Providing access to jar files for OC4J

- 1 Locate the files `Siebel.jar`, `SiebelJI_Common.jar`, and `SiebelJI_enu.jar` in the directory `SIEBEL_HOME\siebsrvr\CLASSES` and copy them to the appropriate directory:
  - For standalone OC4J: `OC4J_HOME/j2ee/home/applib`.
  - For Oracle Application Server: `ORACLE_AS_HOME/j2ee/oc4j_instance/applib`.

For Siebel 7.5.3, all three jars are required. For Siebel 7.7, 7.8, and 8.0, only `SiebelJI.jar` and `SiebelJI_enu.jar` are required.

- 2 Restart OC4J.

#### 6.4.1.2 Providing access to jar files for 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 **Environment** and choose **Shared Libraries**. On the right, click **New**.
- 3 Enter `Siebel Lib` for **Name**, then enter the following values for **Classpath**:

```
SIEBEL_HOME\siebsrvr\CLASSES\SiebelJI.jar;  
SIEBEL_HOME\siebsrvr\CLASSES\SiebelJI_common.jar;  
SIEBEL_HOME\siebsrvr\CLASSES\SiebelJI_enu.jar
```

Make sure to replace `SIEBEL_HOME` with the actual Siebel Server installation path.

Each classpath component must be entered on a separate line. For Siebel 7.5.3, all three jars are required. For Siebel 7.7, 7.8, and 8.0, only `SiebelJI.jar` and `SiebelJI_enu.jar` are required.



If the Siebel Server installation is on a different computer than the WebSphere installation, you must copy these jar files to a location on the WebSphere computer. The Classpath values should point to the jar files on the WebSphere computer.

- 4 Click **OK**.

- 5 In the tree on the left, expand **Applications**, select **Enterprise Applications**, and then select **OracleRTD**.
- 6 At the bottom of the screen, click **Additional Properties**, then **Libraries**. Click **Add**.
- 7 Select **Siebel Lib** from the drop-down list, then click **OK**.
- 8 Click **Save**.


#### 6.4.1.3 Providing access to jar files for WebLogic

- 1 Go to the `BEA_HOME\user_projects\domains\domain_name\bin` and open the `startWebLogic.cmd` file for editing.
- 2 Add the following new lines to the top of the file:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;SIEBEL_HOME\siebsrvr\
CLASSES\SiebelJI.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;SIEBEL_HOME\siebsrvr\
CLASSES\SiebelJI_common.jar;
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;SIEBEL_HOME\siebsrvr\
CLASSES\SiebelJI_enu.jar;
```

Make sure to replace `SIEBEL_HOME` with the actual Siebel Server installation path.

Each classpath component must be entered on a separate line. For Siebel 7.5.3, all three jars are required. For Siebel 7.7, 7.8, and 8.0, only `SiebelJI.jar` and `SiebelJI_enu.jar` are required.



**Note:** If the Siebel Server installation is on a different computer than the WebLogic installation, you must copy these jar files to a location on the WebLogic computer. The Classpath values should point to the jar files on the WebLogic computer.

- 3 Save and close the file.

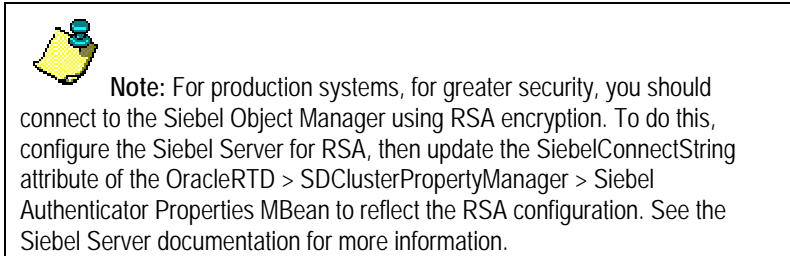
#### 6.4.2 Enabling authentication in Oracle RTD

- 1 Start Oracle RTD.
- 2 Start JConsole and log in. See Section 10.1, *Accessing JConsole*, for more information.
- 3 Go to the **OracleRTD > SDClusterPropertyManager > Security Properties** MBean.
- 4 Set the **AuthenticationEnabled** property to `true`.
- 5 Set the **AuthenticationProviderClass** property to `com.sigmadynamics.server.security.SiebelAuthenticator`.
- 6 Go to the **OracleRTD > SDClusterPropertyManager > Siebel Authenticator Properties** MBean.
- 7 For **SiebelAdminPassword** and **SiebelAdminUser**, enter the credentials of a valid Siebel Server user.



- 8 For **SiebelConnectString**, provide a valid connect string to the Siebel Server. For example:  
`Siebel.TCPIP.None.None://111.22.33.44:2321/Siebel782/SMObjMgr_enc`

Refer to the Siebel Server documentation for more information about the format for the connect string.



When you are using Siebel Object Manager Authentication, use Oracle's Siebel Server to create users and groups for Oracle RTD. Then, assign Oracle RTD permissions to those users and groups using JConsole. See Section 6.5, Assigning permissions to users and groups, for more information.

## 6.5 Assigning permissions to users and groups

Use JConsole to assign permissions to users and groups. Use the method described in the following steps regardless of whether you are using Windows Authentication, Oracle RTD Platform Authentication, or Siebel Object Manager Authentication.

- 1 Start Oracle RTD.
- 2 Start JConsole and log in. See Section 10.1, Accessing JConsole, for more information.
- 3 Go to the **OracleRTD > SDManagement > SecurityManager** MBean.
- 4 Use the `assignPermission()` operation to add permissions to a user or group. When you use the `assignPermission()` operation, you specify the name of a user or group and a permission code. Permission codes include:
  - 0 – Administrator
  - 1 – Open Service
  - 2 – Open Service for Reading
  - 3 – Deploy Service from Studio
  - 4 – Download Service
  - 6 – Execute Integration Points from Studio

You can also use the `listPermissionCodes()` operation to list the available permissions.

**Note:** Permission code 5, Execute Integration Points from Load Generator, is not supported in this release.

## 6.6 Changing the authentication type

Oracle RTD can use Windows Authentication, Oracle RTD Platform Authentication, or Siebel Object Manager Authentication. You can change the type of authentication used by Oracle RTD using JConsole, as follows:

- 1 Start Oracle RTD.
- 2 Start JConsole and log in. See Section 10.1, Accessing JConsole, for more information.

- 3 Go to the **Oracle RTD > SDClusterPropertyManager > Security Properties** MBean.
- 4 Ensure that **AuthenticationEnabled** is set to true.
- 5 Change **AuthenticationProviderClass** as desired:
  - **Windows Authentication:** `com.sigmadynamics.server.security.WindowsAuthenticator`
  - **Oracle RTD Platform Authentication:** `com.sigmadynamics.server.security.DBAuthenticator`
  - **Siebel Object Manager Authentication:** `com.sigmadynamics.server.security.SiebelAuthenticator`

The type of authentication you choose must be configured using the instructions provided in the previous sections.

After you change the authentication type, make sure to assign the appropriate permissions to the users and groups for this authentication type. For more information, see Section 6.5, Assigning permissions to users and groups.

## 7 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 *System Requirements and Supported Platforms for Oracle Real-Time Decisions* for Version 2.2.

This section contains the following topics:

- 7.1 Creating additional JDBC data sources in OC4J
- 7.2 Creating additional JDBC data sources in WebSphere
- 7.3 Configuring additional JDBC data sources in WebLogic
- 7.4 Testing a new enterprise data source

### 7.1 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.

#### 7.1.1 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.

- 1 If you are using standalone OC4J, copy the appropriate jar files to `OC4J_HOME/j2ee/home/app/lib`. If you are using Oracle Application Server, copy the appropriate jar files to `ORACLE_AS_HOME/j2ee/oc4j_instance/app/lib`. Copy the jars appropriate for your data source:
  - a. For **SQL Server**, copy the file `sqljdbc.jar` in the directory `RTD_HOME/lib/jdbc`.
  - b. For **DB2**, copy the files `db2jcc.jar` and `db2jcc_license_cu.jar` in `RTD_HOME/lib/jdbc`.
  - c. For **Teradata**, copy the files `terajdbc4.jar`, `tdgssjava.jar`, and `tdgssconfig.jar`.  
  
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.
  - d. 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.

## 7.1.2 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.

- 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.

**Note:** 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**.

**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.driver.OracleDriver`

**Note:** If you are adding an Oracle Database data source, you *must* enter `oracle.jdbc.driver.OracleDriver` for **Connection Factory Class**. Do not use the default value of `oracle.jdbc.pool.OracleDataSource`.

- **DB2:** `com.ibm.db2.jcc.DB2Driver`
- **Teradata:** `com.ncr.teradata.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`
- **Oracle Database:** `jdbc:oracle:thin:@db_host:db_port:sid`
- **DB2:** `jdbc:db2://db_host:db_port/db_name`
- **Teradata:** `jdbc:teradata://db_host/DATABASE=db_name`
- **Oracle BI EE:** `jdbc:oraclebi://server_name:9703/user=bi_user_name;password=bi_password;catalog=catalog_name;`

**Note:** 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 Containers for J2EE 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 that `default` is selected for **Application**, and that `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, DB2, or Teradata 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**.

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.

### 7.1.3 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.

- 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.

- a. Locate the file `./rtis/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_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 7.1.2. 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.

- b. Locate the file `./soap/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 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 7.1.2. 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.

- 2 Go to the directory `OC4J_HOME/j2ee/home/application-deployments/OraclERTD`. Make sure to go to the `application-deployments` directory, not the `applications` directory.

- a. Locate the file `./rtis/WEB-INF/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 7.1.2. For example:

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

Then, save the changes and close the file.

- b. Locate the file `./soap/WEB-INF/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 7.1.2. For example:

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

Then, save the changes and close the file.

- 3 Start OC4J.

## 7.2 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.

### 7.2.1 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.

- 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\_nameNode\_number**), then click **New**.
- 5 For **Database type**, select **User-defined**.
- 6 For **Implementation class name**, enter one of the following values:
  - a. **SQL Server:** `com.microsoft.sqlserver.jdbc.SQLServerConnectionPoolDataSource`
  - b. **Oracle Database:** `oracle.jdbc.pool.OracleConnectionPoolDataSource`
  - c. **DB2:** `com.ibm.db2.jcc.DB2ConnectionPoolDataSource`
  - d. **Teradata:** `com.ncr.teradata.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:
  - a. **SQL Server:** `RTD_HOME/lib/jdbc/sqljdbc.jar`
  - b. **Oracle Database:** `RTD_HOME/lib/jdbc/ojdbc14.jar`
  - c. **DB2:** `RTD_HOME/lib/jdbc/db2jcc.jar;RTD_HOME/lib/jdbc/db2jcc_license_cu.jar`
  - d. **Teradata:** `RTD_HOME/lib/jdbc/terajdbc4.jar;RTD_HOME/lib/jdbc/tdgssjava.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**. Then, click **OK**.
  - f. 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**.

### 7.2.2 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: `WEBSPPHERE_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 7.2.1.

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 When you have completed the new entry, save and close the file.
- 4 Open the following file for editing: `WEBSPPHERE_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 7.2.1.

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

### 7.2.3 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.

- 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 7.2.1.
- 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 7.2.1.
- 7 Click **Apply**.
- 8 Scroll to the bottom of the table and click **OK**, then click **Save**.

### 7.2.4 Adding additional data sources

If you want to add another data source of the same type, start from Step 10 of Section 7.2.1. 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.

## 7.3 **Configuring 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.

### 7.3.1 Setting the path to JDBC jar files for your data source

- 1 If the enterprise data source you want to add is a Teradata data source, 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 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 If WebLogic is running on Windows, follow these steps to set the classpath to include the path to the JDBC jar files for your data source:
  - a. Open the file `BEA_HOME\user_projects\domains\domain_name\bin\startWebLogic.cmd` for editing.
  - b. At the top of the file, insert the appropriate lines to set the classpath to include the path 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.

For SQL Server, include the following additional line:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\sqljdbc.jar;
```

For Oracle Database, include the following additional line:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\ojdbc14.jar;
```

For DB2, include the following two additional lines:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\db2jcc.jar;  
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\db2jcc_license_cu.jar;
```

For Teradata, include the following three additional lines:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\terajdbc4.jar;  
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\tdgssjava.jar;  
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\tdgssconfig.jar;
```

For Oracle BI EE, include the following additional line:

```
set EXT_PRE_CLASSPATH=%EXT_PRE_CLASSPATH%;RTD_HOME\lib\jdbc\bijdbc.jar;
```

Make sure to replace `RTD_HOME` with the actual Oracle RTD installation path.

**Note:** You do not need to insert any lines if you are adding a Siebel Analytics Server data source.

- c. Save and close the file.

- d. Restart WebLogic.
- 4 If WebLogic is running on UNIX, follow these steps to set the classpath to include the path to the JDBC jar files for your data source.
- a. Open the file `BEA_HOME/user_projects/domains/domain_name/bin/startWebLogic.sh` for editing.
  - b. At the top of the file, insert the appropriate lines to set the classpath to include the path 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.

For SQL Server, include the following additional line:

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/sqljdbc.jar"
```

For Oracle Database, include the following additional line:

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/ojdbc14.jar"
```

For DB2, include the following two additional lines:

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc.jar"
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/db2jcc_license_cu.jar"
```

For Teradata, include the following three additional lines:

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/terajdbc4.jar"
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/tdgssjava.jar"
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/tdgssconfig.jar"
```

For Oracle BI EE, include the following additional line

```
EXT_PRE_CLASSPATH="${EXT_PRE_CLASSPATH}:RTD_HOME/lib/jdbc/bijdbc.jar"
```

Make sure to replace `RTD_HOME` with the actual Oracle RTD installation path.

**Note:** You do not need to insert any lines if you are adding a Siebel Analytics Server data source.

- c. Save and close the file.
- d. Restart WebLogic.

### 7.3.2 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.

- 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 > BEA Products > User Projects > *domain\_name* > Admin Server Console**.

- 2 In the tree on the left, expand **Services**, then expand **JDBC** and choose **Data Sources**.
- 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.driver.OracleDriver`
  - **DB2:** `com.ibm.db2.jcc.DB2Driver`
  - **Teradata:** `com.ncr.teradata.TeraDriver`
  - **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`
- **Oracle Database:** `jdbc:oracle:thin:@db_host:db_port:sid`
- **DB2:** `jdbc:db2://db_host:db_port/db_name`
- **Teradata:** `jdbc:teradata://db_host/DATABASE=db_name`
- **Oracle BI EE:** `jdbc:oraclebi://server_name:9703/user=bi_user_name;password=bi_password;catalog=catalog_name;`

**Note:** 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, **AdminServer**). You *must* perform this step before completing the data source configuration.
- 13 Click **Finish**.
- 14 Click **Activate Changes**.

### 7.3.3 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 Go to the directory where you expanded the `RTD.ear` file during installation (`RTD_HOME/package/expanded`).
- 2 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 7.3.2. 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`.

- 3 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 7.3.2. 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`.

## 7.4 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.

- 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.

## 8 Additional Configuration Settings and Starting Client Tools

This section describes additional configuration settings for Decision Center and Real-Time Decision Server.

This section contains the following topics:

- 8.1 Decision Center Browser Configuration
- 8.2 Real-Time Decision Server Logging Configuration
- 8.3 Accessing Oracle Real-Time Decisions client tools

### 8.1 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.

### 8.2 Real-Time Decision Server Logging Configuration

The **Logger** attribute of the **OracleRTD > SDManagement > SDLogging** MBean is a Log4J-type logger. 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 10.1, Accessing JConsole, for more information.
- 3 Go to the **Oracle RTD > SDManagement > SDLogging** MBean.
- 4 Change the logging attributes as necessary:
  - a. 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.
  - b. 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.

### 8.3 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. For information about starting the JConsole administration tool, see Section 10.1, Accessing JConsole.

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.



### **8.3.1 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`.

### **8.3.2 Accessing Decision Center**

To access Decision Center from any computer, open a Web browser and go to `http://server_name:8080/ui`.

### **8.3.3 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`.

## 9 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 section contains the following topics:

- 9.1 All-In-One configuration
- 9.2 Full Deployment configuration

### 9.1 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.

Inline Intelligence™ Platform	Minimum Configuration
J2EE Application Server	OC4J
OS	Windows
Number of CPUs	2
CPU	2+ GHz
RAM	2 GB
HD	100 GB
Database	SQL Server, Oracle Database, or DB2

### 9.2 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.

Server	Component	Minimum Configuration
Decision Servers (2)	CPU	2 X 2-CPU, 2+GHz, 2GB RAM
	J2EE Application Server	OC4J, WebSphere, or WebLogic
	OS	Windows, Solaris, AIX, HP-UX, or Linux
Database Server <sup>1</sup>	CPU	4-CPU, 2+GHz, 8GB RAM
	DBMS	SQL Server, Oracle Database, or DB2
Learning Server <sup>2</sup>	CPU	2-CPU, 2+GHz, 2GB RAM
	J2EE Application Server	OC4J, WebSphere, or WebLogic
	OS	Windows, Solaris, AIX, HP-UX, or Linux

Server	Component	Minimum Configuration
<p>Notes:</p> <ol style="list-style-type: none"><li>1: The ratio of database server CPUs to Decision Server CPUs is approximately 1:1.</li><li>2: The ratio of Learning Server CPUs to Decision Server CPUs is approximately 1:4.</li></ol>		

## 10 JMX Management of Oracle Real-Time Decisions

Oracle Real-Time Decisions uses the J2EE industry standard Java Management Extensions (JMX) to manage the Management Service, Decision Service, Learning Service, and deployed Inline Services. You can use JConsole, a JMX management tool that comes with Java Development Kit 1.5.0.8, to manage Oracle RTD. Third-party JMX implementations may also be used to manage Oracle RTD. This document assumes you are using JConsole.

JMX MBeans manage various aspects of Oracle RTD, including logging, Inline Service configuration, security, and users. Oracle RTD is comprised of three Services:

**Management Service:** Manages the generation of Inline Services from configurations that are created using Decision Studio.

**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 section contains the following topics:

- 10.1 Accessing JConsole
- 10.2 About JMX MBean operations and attributes
- 10.3 Oracle Real-Time Decisions cluster-level management
- 10.4 Oracle Real-Time Decisions member-level management
- 10.5 Oracle Real-Time Decisions security management
- 10.6 Managing Inline Services
- 10.7 Managing Deployment States
- 10.8 Managing the Learning Service
- 10.9 Managing the Alert Service

### 10.1 Accessing JConsole

To access JConsole for OC4J, Oracle Application Server, or WebLogic, follow these steps:

- 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, then enter the following:
  - a. For **Host**, enter the name of the computer where Real-Time Decision Server is running.
  - b. For **Port**, enter your JMX remote port number (for example, 12345).
  - c. For **User Name** and **Password**, enter the credentials you specified when you set up JConsole. See Section 3.7, *Setting up JConsole for OC4J*, and Section 5.7, *Setting up JConsole for WebLogic*, for more information.

To access JConsole for WebSphere, follow these steps:

- 1 On the computer where WebSphere is installed, run the batch script you created in Section 4.7.2, *Setting up a batch file for JConsole*.
- 2 Log in as the JConsole user you created in Section 4.7.3, *Creating a JConsole user* (for example, `jmx_admin`).



**Note:** When Oracle RTD is running on WebSphere, JConsole must be run on the same computer as the WebSphere server, by default. This problem occurs because of WebSphere limitations.

## 10.2 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.
- **AlertService** – Administers the Alerts Service.

## 10.3 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.

### 10.3.1 About OracleRTD > SDManagement > SDClusterPropertyManager

The SDManagement > SDClusterPropertyManager MBean has the following operation:

`java.lang.Void restoreDefault()`

**Parameters:** none

**Purpose:** Restores the default installation settings to Oracle RTD. If this command is run on a cluster, then values are restored to cluster defaults.

### 10.3.2 About OracleRTD > SDClusterPropertyManager > Misc

The SDClusterPropertyManager > Misc MBean has the following attributes:

Attribute	Description
DisableBatchDBOperations	Boolean switch that controls batch database operations.
AutoFlushTimeout	Interval in seconds controlling auto flush of database write buffers. Fractional values are supported.
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.
IntegrationPointRequestIOFactor	The percentage of time Integration Point requests spend doing IO, or otherwise waiting for systems external to this VM.
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". $\text{NumCPUs} * \text{Math.ceil}(1/(1-\text{DSRequestIOFactor})) + 5$
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.
IntegrationPointGuaranteedRequestTimeout	Guaranteed response time, in milliseconds, for Integration Point requests. (Service Level Guarantee). Zero means don't timeout Integration Point requests - suitable for debugging only.

Attribute	Description
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.
SystemDSName	The JNDI name of the datasource
ModelDSName	The JNDI name of the datasource used by the Learning Service
ArchivedModelCatalogRefreshInterval	Refresh interval in seconds for a catalog of archived models. The catalog is used by Discovery Explorer.
ArchivedModelCacheTimeToLive	Maximum time in seconds an archived model is preserved in memory. The cache of archived models is used by Discovery Explorer.
DSQueueThreadCount	The number of JMS Threads listening to the DS request queue
DSSessionIdleTimeoutMilliSec	Decision Service session idle timeout in milliseconds.
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.
CustomLoginLogo	Custom image for login page
CustomHeaderLogo	Custom image for main page
SMTPHost	SMTP Mail server host name. Currently only used to send alert notifications.
SMTPPort	SMTP Mail server port.
SMTPUser	SMTP Mail server user name, if SMTP authorization is required.
SMTPPassword	SMTP Mail server password, if SMTP authorization is required.

### 10.3.3 About OracleRTD > SDClusterPropertyManager > Cluster

The SDClusterPropertyManager > Cluster MBean has the following attributes:

Attribute	Description
GeneratedDSCookies	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.

Attribute	Description
JGroupProtocols	Defines the protocols and properties used for the JGroups channel.
JGroupDSProtocols	Defines the JGroups protocols and properties used for the Decision Server forwarding channel.
SDGroupName	This is the name of the cluster.
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 machine fails or leaves the cluster before trying to restart the AlertService.
AlertServiceInitialWait	The number of milliseconds to wait when the server first starts up before trying to start the Alert Service.
AlertServiceRestartWait	The number of milliseconds to wait after a machine fails or leaves the cluster before trying to restart the Alert Service.
OperationalDataCleanupPeriod	The number of hours (fractions are allowed) between cleanup of the operational data (choice history, statistics, learning data storage) in the database.
StatisticsCleanupChunkSize	The chunk size to use when deleting old statistic records.
ChoiceHistoryCleanupChunkSize	The chunk size to use when deleting old choice history records.
LearningDataStorageCleanupChunkSize	The chunk size to use when deleting old learning data storage records.
StatisticsCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.
ChoiceHistoryCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.
LearningDataStorageCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.

#### 10.3.4 About OracleRTD > SDClusterPropertyManager > Deployment

The SDClusterPropertyManager > Deployment MBean has the following attributes:

Attribute	Description
AppPollingInterval	How frequently, in seconds, the AppFactory polls the SDApps table to see if there are new apps.



### 10.3.5 About Oracle RTD > SDClusterPropertyManager > Alert Service

The SDClusterPropertyManager > Alert Service MBean has the following attributes:

Attribute	Description
MailFrom	Alert email notifications will be sent from this address.
MailLogo	Alert email notifications will have this logo in their header.
AlertHistoryCleanupIntervalMinutes	The Alert History archive should be scanned at this interval for expiration of old alerts. Units are minutes.
AlertDispatcherSleepIntervalMinutes	The Alert Service should evaluate alerts at this interval. Units are minutes.
AlertDispatcherErrorSleepIntervalMinutes	The Alert Service should fall back to evaluating alerts at this interval if there are any errors. Units are minutes.

## 10.4 Oracle Real-Time Decisions member-level management

### 10.4.1 About OracleRTD > SDManagement > SDLogging

The SDManagement > SDLogging MBean has the following attributes:

Attribute	Description
Priority	Current logging priority for the <code>com.sigmadynamics</code> category. Valid values are <code>DEBUG</code> , <code>ERROR</code> , <code>INFO</code> , and <code>WARN</code> .
InlineServicePriority	Current logging priority for Inline Services. Valid values are <code>DEBUG</code> , <code>ERROR</code> , <code>INFO</code> , and <code>WARN</code> .
LogFile	Location of the current log file.

For information about the values written to the log file, see *Decision Studio Reference Guide*, Section 5: Troubleshooting and Debugging Inline Services.

### 10.4.2 About OracleRTD > SDPropertyManager > Performance Monitoring

The SDPropertyManager > Performance Monitoring MBean has the following attributes:

Attribute	Description
DSPerfCounterEnabled	Enables the writing of DS performance counters. This 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.

Attribute	Description
DSPerfCounterLogFile	The tab-separated CSV file into which DS performance counts are periodically appended. If MS Excel is available, <code>ds_perf.xls</code> , supplied in the installation's <code>etc</code> directory, provides a convenient view.
DSPerfCounterLogInterval	The update interval in milliseconds for DS performance counts.

For more about using Performance Monitoring, see *Decision Studio Reference Guide: Using Performance Monitoring*.

#### 10.4.3 About OracleRTD > SDPropertyManager > Misc

The SDPropertyManager > Misc MBean has the following attributes:

Attribute	Description
DecisionServiceEnabled	Whether or not Decision Service should run in this instance.
LearningServiceEnabled	Whether or not Learning Service should run in this instance.
AlertServiceEnabled	Whether or not Alert 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.
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.
IntegrationPointRequestIOFactor	The percentage of time Integration Point requests spend doing IO, or otherwise waiting for systems external to this VM.
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 <code>Math.ceil</code> means "round up to the next higher integer". $\text{NumCPUs} * \text{Math.ceil}(1/(1 - \text{DSRequestIOFactor})) + 5$
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 <code>IntegrationPointMaxConcurrentJobs</code> . The calculated value of <code>IntegrationPointMaxConcurrentJobs</code> can be seen in <code>SDConsole/Members/Decision Service</code> .
WebServerPort	The port of the Web Server. Requires server restart before the property change takes effect. This port is used for internal communications between multiple RTD servers.

Attribute	Description
HTTPSEnabled	Whether or not HTTPS is enabled on the Web Server port. Requires server restart before the property change takes effect.

#### 10.4.4 About OracleRTD > Server > DecisionService

The Server > DecisionService MBean has the following read-only attributes:

Attribute	Description
MaxAllowedConcurrentRequests	Maximum number of requests that can run concurrently.
CurrentRequestsRunning	Number of currently running requests.
CurrentRequestsQueued	Number of currently waiting requests, not yet running.
TotalRequests	Total number of requests seen since the server started.
TimedOutRequests	Total number of requests that have timed out.
RequestsForwarded	Total number of requests forwarded to another server in the cluster.
RequestQueueCapacity	Maximum number of requests that can wait concurrently.
RequestsQueued	Total number of requests that have had to wait before running.
PeakRequestsQueued	Largest number of requests that have had to wait at any one time.
RequestsWhenQueueFull	Total number of requests rejected because the request queue was full.
CurrentSessions	Number of Decision Service sessions still open.
TotalSessions	Total number of Decision Service sessions created.

## 10.5 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.

### 10.5.1 About OracleRTD > SDManagement > SecurityManager

The SDManagement > SecurityManager MBean has the following operations:

#### `addGroupMember()`

**Parameters:** group Name: The name of the group to which you want to add members.  
username: The name of the user you want to add.

**Purpose:** Makes a user or group a direct member of a group.

### **assignPermission()**

**Parameters:** userOrGroup: The user name or group name to which you want to assign permissions.  
permCode: The code of the permission.

**Purpose:** Assigns the specified cluster permission. See Section 6.5, Assigning permissions to users and groups for more information.

### **createGroup()**

**Parameters:** groupName: The name of the group you want to create.  
description: Description for the new group.

**Purpose:** Creates a new security group. Oracle RTD Platform Authentication only.

### **createUser()**

**Parameters:** username: The name of the user you want to create.  
description: A description for the new user.  
password: The password for the user, in clear text.

**Purpose:** Creates a new user. Oracle RTD Platform Authentication only.

### **deleteGroup()**

**Parameters:** groupName: The name of the group you want to delete.

**Purpose:** Deletes a security group. Oracle RTD Platform Authentication and Siebel Object Manager Authorization only.

### **deleteUser()**

**Parameters:** username: The name of the user you want to delete.

**Purpose:** Deletes an existing user. Oracle RTD Platform Authentication only.

### **listDirectGroupMembers()**

**Parameters:** groupName: The name of the target group.  
includeGroups: True or False. If true, include member groups, otherwise include only users.

**Purpose:** Returns the names of users and groups directly as a member of a specified group.

### **listDirectlyContainingGroups ()**

**Parameters:** userOrGroup: The target user or group name.

**Purpose:** Returns the names of groups containing a specified user or group as a direct member.

### **listDirectPermissions()**

**Parameters:** userOrGroup: The target user name or group name.

**Purpose:** Returns the permissions directly assigned to a specific user or group.

### **listEffectivePermissions()**

**Parameters:** userOrGroup: The target user name or group name.

**Purpose:** Returns the permissions directly assigned to a specific user or group, or to any of its directly or indirectly containing groups.

### **listEveryoneHavingDirectPermissions()**

**Parameters:** none

**Purpose:** Returns the names of users and groups that have directly assigned permissions.

### **listGroupNames()**

**Parameters:** none

**Purpose:** Returns the names of all groups to which permissions may be assigned.

### **listPermissionCodes()**

**Parameters:** none

**Purpose:** 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.

### **listUserNames()**

**Parameters:** none

**Purpose:** Returns the names of all users to which permissions may be assigned.

### **removeAllDirectPermissions ()**

**Parameters:** userOrGroup: The target user name or group name

**Purpose:** Removes all permissions directly assigned for the specified user or group.

### **removeGroupMember()**

**Parameters:** containingGroupName: The group to contain the member group.  
memberGroupName: The group to be added as a member of another group.

**Purpose:** Removes a user or group from a containing group.

### **removePermission()**

**Parameters:** userOrGroup: The target user name or group name.  
permCode: The code of the permission you want to remove.

**Purpose:** Removes a permission from its direct assignment to a user or group. The permission is not removed from any groups containing the specified user or group.

### **renameGroup()**

**Parameters:** oldGroupName: The previous name of the group.  
newGroupName: The new name of the group.

**Purpose:** Renames a group.

renameUser()

**Parameters:** oldUserName: The previous name of the user.  
newUserName: The new name of the user.

**Purpose:** Renames a user.

setPassword()

**Parameters:** username: The name of the target user.  
password: The new password, in clear text.

**Purpose:** Changes the password for a given user.

### 10.5.2 About OracleRTD > SDClusterPropertyManager > Security Properties

The SDClusterPropertyManager > Security Properties MBean has the following attributes:

Attribute	Description
AuthenticationEnabled	When false, password checking and permission checking is turned off.
AuthenticatorLogsInvalidCredentials	When true, the Authenticator logs a short message whenever a login failure is believed to have been caused by invalid credentials.
AuthenticatorLogsFailures	When true, the Authenticator logs details about each failed login, including an exception stack trace. This level of detail is provided unconditionally for errors deemed by the security provider to be caused by configuration problems; this flag provides similar details for errors that the provider thinks, perhaps incorrectly, are caused by invalid credentials.
AuthenticationProviderClass	The name of the class that performs logins, group membership queries, and optionally manages the definition of users and groups. Use <code>com.sigmadynamics.server.security.WindowsAuthenticator</code> , <code>com.sigmadynamics.server.security.DBAuthenticator</code> , or <code>com.sigmadynamics.server.security.SiebelAuthenticator</code> .

### 10.5.3 About OracleRTD > SDManagement > SecurityAuthenticator

The SDManagement > SecurityAuthenticator MBean has the following read-only attributes:

Attribute	Description
SecurityManager	Security Manager
ProviderClass	Authentication provider class.

The SDManagement > Security Authenticator MBean has the following operations:

### java.lang.String loginUserPassword()

**Parameters:** username: The name of the user.  
password: The password of the user, in clear text.

**Purpose:** Log in with a user name and password, to receive a login ticket.

### java.lang.String getUserName()

**Parameters:** ticket: The ticket returned from the login.

**Purpose:** Get the user name associated with the login session.

### java.lang.Boolean isUserInRole()

**Parameters:** ticket: The ticket returned from the login.  
userOrGroupName: The name of the user or group being tested for identity or containment, respectively.

**Purpose:** Returns true if the logged-in user is the specified user, or is directly or indirectly in the specified group.

### java.lang.Void logout()

**Parameters:** ticket: The ticket returned from the login.

**Purpose:** Terminates a login session.

## 10.6 Managing Inline Services

Use Inline Service Manager to manage the Inline Services deployed on the cluster.

### 10.6.1 About OracleRTD > SDManagement > InlineServiceManager

Each deployed Inline Service is displayed under the InlineServiceManager MBean.

The SDManagement > InlineServiceManager MBean has the following operation:

#### removeAllServices ()

**Parameters:** none

**Purpose:** Removes all inline services (loaded, loadable, failed).

### 10.6.2 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
ServiceId	Service id

Attribute	Description
DeploymentState	Deployment state
Flag	Flag
LockStatus	Lock status

The InlineServiceManager MBeans for a specific Inline Service have the following operations:

Operation name	Parameters	Purpose
unlockService()	none	Unlocks this service.
removeService()	none	Stops an inline service in this server and removes the service from the database.
flushStatistics()	none	Flushes all of the statistics for this service to the database.
deleteStatistics()	none	Flushes and deletes all of the statistics for this service from the database.
deleteChoiceHistory()	none	Deletes all of the choice history for this service from the database.
deleteAllOperationalData()	none	Deletes all of the operational data for this service from the database. This includes choice history, statistics, and the study.
deleteStudy()	none	Removes the study for this service

## 10.7 Managing Deployment States

By viewing the OracleRTD > DeploymentStates MBeans, you can see a list of deployment states that are available on the cluster. The OracleRTD > SDManagement > DeploymentStates MBeans allow you to add, edit, delete, and reorder deployment states.

### 10.7.1 About OracleRTD > SDManagement > DeploymentStates

The SDManagement > DeploymentStates MBean has the following attributes:

Attribute	Description
StateObjectNames	A listing of all deployment states available on the server.

The SDManagement > Deployment States MBean has the following operations:



## addDeploymentState ()

**Parameters:** Name - Deployment state name.  
AfterDeploymentState - Name of the deployment state after which the new one should be created (if no value is passed in, the new deployment state will be created at the beginning of the list).  
AllowHotSwapping - Allows hot swapping of Inline Services. If Hot Swapping is enabled for this deployment state, and an Inline Service is re-deployed in this 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.

**Purpose:** Creates a new deployment state.

### 10.7.2 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
Name	Name of the deployment state.
Id	Id of the deployment state.
AllowHotSwapping	Allow hot swapping of Inline Services with this deployment state in Decision Service.

Each Deployment States > [State] MBean has the following operations:

## remove ()

**Parameters:** None.

**Purpose:** Removes a deployment state from the cluster.

## 10.8 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.

### 10.8.1 About OracleRTD > Server > LearningService

The Server > LearningService MBean has the following attributes:

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.

### 10.8.2 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 operations:

#### Delete()

**Parameters:** None

**Purpose:** Deletes this study.

### 10.8.3 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()

**Parameters:** None

**Purpose:** Deletes this model.

#### DeleteAttributeValue()

**Parameters:** AttributeName - The name of an attribute.  
Value - The value to be deleted.

**Purpose:** Erases model data collected for a value of an attribute.

#### DeleteAttributeValueRange()

**Parameters:** AttributeName - The name of an attribute.  
HighestValue - The highest value to be deleted.  
LowestValue - The lowest value to be deleted.

**Purpose:** Erases model data collected for a range of values of an attribute.

StartNewTimeWindow()

**Parameters:** None

**Purpose:** Closes the current time window and starts a new one. This operation should not be used in production environment since it may impair future model learning.

## 10.9 Managing the Alert Service

You can use the `SDClusterPropertyManager > Alert Service MBean` to adjust parameters related to the Alert Service.

### 10.9.1 About OracleRTD > SDClusterPropertyManager > Alert Service

The `SDClusterPropertyManager > Alert Service MBean` has the following attributes:

Attribute	Description
MailFrom	Alert email notifications will be sent from this address.
MailLogo	Alert email notifications will have this logo in their header.
AlertHistoryCleanupIntervalMinutes	The Alert History archive should be scanned at this interval for expiration of old alerts. Units are minutes.
AlertDispatcherSleepIntervalMinutes	The Alert Service should evaluate alerts at this interval. Units are minutes.
AlertDispatcherErrorSleepIntervalMinutes	The Alert Service should fall back to evaluating alerts at this interval if there are any errors. Units are minutes.