Oracle® Fusion Middleware

Administrator's Guide for Oracle Real-Time Decisions

11*g* Release 1 (11.1.1)

E16632-04

February 2013

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

Oracle Fusion Middleware Administrator's Guide for Oracle Real-Time Decisions, 11*g* Release 1 (11.1.1)

E16632-04

Primary Author: Oracle Corporation

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

Contents

5 Configuring Data Access for Oracle Real-Time Decisions

6 Clustering and High Availability for Oracle Real-Time Decisions

7 Additional Configuration Settings and Starting Client Tools

8 Production Deployment of Oracle Real-Time Decisions

9 Command Line Deployment of Inline Services

10 Setting Up and Using Model Snapshots

11 Performance Monitoring

12 Backup and Recovery of Oracle Real-Time Decisions

13 Managing Oracle Real-Time Decisions

A System Log and Configuration Files

B Upgrading and Patching Oracle Real-Time Decisions

Preface

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

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

Audience

This document is intended for administrators of Oracle RTD. Oracle RTD administrators should have a working knowledge of how to administer enterprise-level applications.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Fusion Middleware 11*g* Release 1 (11.1.1) documentation set:

- *Oracle Fusion Middleware Release Notes* for your platform
- *Oracle Fusion Middleware Decision Center User's Guide for Oracle Real-Time Decisions*
- *Oracle Fusion Middleware Platform Developer's Guide for Oracle Real-Time Decisions*
- *Oracle Fusion Middleware Administrator's Guide*
- *Oracle Fusion Middleware Application Security Guide*

- *Oracle Fusion Middleware High Availability Guide*
- *Oracle Fusion Middleware Information Roadmap for Oracle WebLogic Server*
- *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*
- *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*
- *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*
- *Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*
- *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*
- *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*
- *Oracle Fusion Middleware Oracle WebLogic Scripting Tool*
- *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

A Note About Path Names

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

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

General Note: When you copy and paste multi-line code or command text from the PDF version of this manual, extra carriage return characters get added to the output, and will cause errors if they are not removed. This problem does not occur in the HTML version. So, if you want to copy and paste multi-line code or command text, do so from the HTML version.

New Features for Oracle RTD Administrators

This preface describes additional system administration features for Oracle RTD Release 11.1.1.7.0.

New System Administration Features for Oracle RTD Release 11.1.1.7.0

New system administration features in Oracle RTD Release 11.1.1.7.0 include the following:

- Ability to install and configure Oracle RTD on IBM WebSphere

  For information on installing, configuring, and administering Oracle RTD on IBM WebSphere, see the chapter "Managing Oracle Business Intelligence on IBM WebSphere" in *Oracle Fusion Middleware Third-Party Application Server Guide*.

1

Introduction to Oracle Real-Time Decisions Administration

This chapter introduces system administration in Oracle Real-Time Decisions (Oracle RTD), describes components, typical system administration tasks, and available system administration tools. It also lists related topics covered in other books, and provides information about system requirements and certification.

> **Note:** This guide provides instructions on administering Oracle RTD on Oracle WebLogic Server.
>
> For information on installing, configuring, and administering Oracle RTD on IBM WebSphere, see the chapter "Managing Oracle Business Intelligence on IBM WebSphere" in *Oracle Fusion Middleware Third-Party Application Server Guide*.

This chapter contains the following topics:

- Section 1.1, "Oracle Real-Time Decisions Components"
- Section 1.2, "What is System Administration in Oracle Real-Time Decisions?"
- Section 1.3, "Topics of Interest in Other Guides"
- Section 1.4, "System Requirements and Certification"

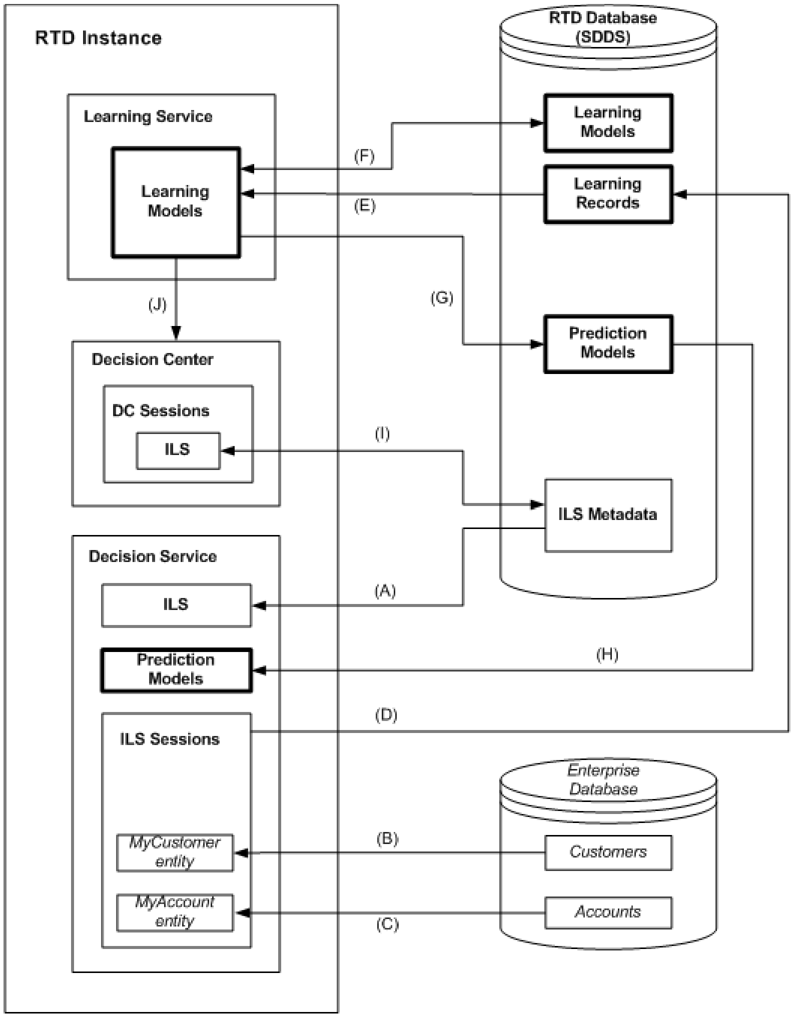
1.1 Oracle Real-Time Decisions Components

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

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

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

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



The main Oracle Real-Time Decisions services are:

- Decision Service

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

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

- Learning Service

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

- Decision Center

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

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

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

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

In more detail:

1. By default, Learning Service wakes up every 15 seconds, and checks to see if there are any new learning records in the database. If so, it retrieves the new records, typically in batches of 1000, and creates or updates the learning models in Learning Service memory.
2. Each model has a persistence interval, which controls how often learning models are written to the database. If the persistence interval is reached or exceeded since the last write-out to the database, Learning Service writes out the learning models and deletes the old learning records.
3. Learning records and learning models contain all the attributes (and their values) defined for the session. The key part of the learning process is selecting those attributes and values that are predictive, that is, which correlate with the output that you want to predict, and generating prediction models from them. In general, prediction models are much smaller than learning models, as only a small proportion of attributes are predictive.
4. You can select whether or not to generate a prediction model for an Oracle RTD model. If predictive modeling is selected, then another model parameter controls when to generate the prediction model. This parameter specifies what percentage of new learning records added to the model will prompt the creation of a new prediction model. The new prediction model replaces the old prediction model.

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

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

Time Windows

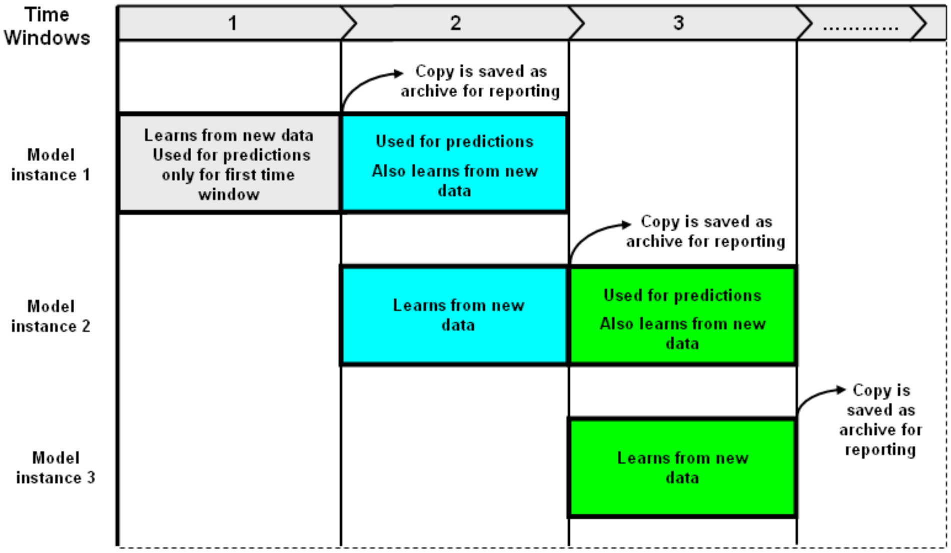
The Time Window parameter also influences how models are processed and used. Time Window defines a time period, such as Week, Month, Quarter, Year, which is defined (and can be altered) at the model level.

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

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

Figure 1–2 shows the model processes as they occur in the initial time windows of a model, and shows how Oracle RTD creates new, overlapping instances of models as time passes.

Figure 1–2 Model Processes in Initial Time Windows


Time Windows
1
2
3
…………
Copy is saved as archive for reporting
Model instance 1
Learns from new data Used for predictions only for first time window
Used for predictions
Also learns from new data
Copy is saved as archive for reporting
Model instance 2
Learns from new data
Used for predictions
Also learns from new data
Copy is saved as archive for reporting
Model instance 3
Learns from new data

Each model instance starts at the beginning of a time window, exists for two time windows, and, apart from the first model instance, overlaps with the previous model instance.

For each model instance, learning information accumulates during its first time window, at the end of which time a copy is saved as archive for reporting. At that point in time:

1. The model instance becomes the "primary model instance", and during its second time window it is used for predictions, while also continuing to learn

2. The next model instance is created, which serves as the "secondary model instance" and is only used for learning purposes until it becomes the primary model itself at the next time window boundary

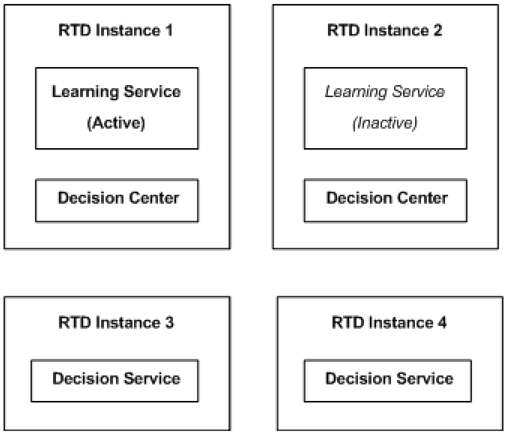
Summarizing:

- Predictions derive from the learning records received so far in the current time window together with all the learning records in the previous time window.
- Apart from the first time window, predictions only derive from a primary model instance, during the second time window of its existence.

Clustering

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

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



- In a cluster, Decision Center is typically deployed in the same Oracle RTD instances as Learning Service, because it needs to talk to Learning Service. Decision Center is not a singleton, however, so any instances not co-located with the active Learning Service can communicate with the active Learning Service.
- In a cluster, Decision Center is typically deployed in a different Oracle RTD instance from Decision Service, to minimize the performance and memory impact of Decision Center on Decision Service.

For more information on Oracle RTD component cluster configurations, see the section "High Availability for Oracle Real-Time Decisions" in *Oracle Fusion Middleware High Availability Guide*.

1.2 What is System Administration in Oracle Real-Time Decisions?

Oracle Real-Time Decisions system administration is described in the following sections:

- Section 1.2.1, "Typical Oracle Real-Time Decisions System Administration Tasks"
- Section 1.2.2, "Oracle Real-Time Decisions as an Oracle Business Intelligence Component"
- Section 1.2.3, "Oracle Real-Time Decisions Configured in a WebLogic Domain"

- Section 1.2.4, "Which System Administration Tools Enable You to Manage Oracle RTD?"

1.2.1 Typical Oracle Real-Time Decisions System Administration Tasks

Table 1–1 shows the typical Oracle RTD system administration tasks that you perform, and indicates where to find related information.

Table 1–1 Oracle Real-Time Decisions System Administration Tasks

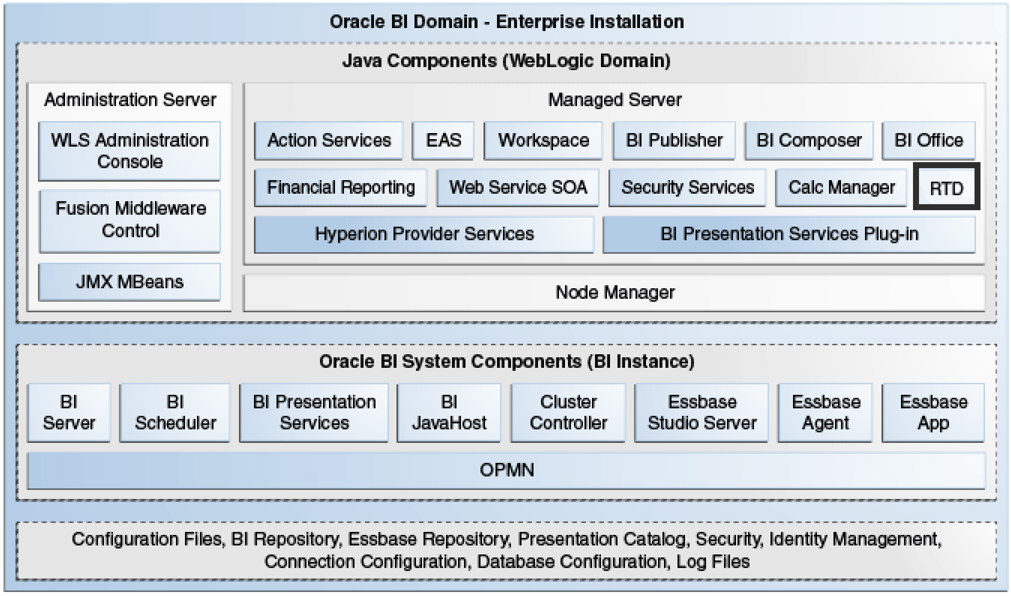
System Administration Task	More Information
Viewing Oracle Real-Time Decisions status	■ Section 2.1.1, "Logging into Fusion Middleware Control"
Starting and stopping Oracle Real-Time Decisions	■ Section 2.3.2, "Starting and Stopping Oracle RTD"
Configuring Oracle Real-Time Decisions	■ Section 3.3, "Configuring Oracle RTD After Installation" and Chapter 5, "Configuring Data Access for Oracle Real-Time Decisions"
Securing the system	■ Chapter 4, "Security for Oracle Real-Time Decisions"
Diagnosing problems and resolving issues	■ Chapter 11, "Performance Monitoring" ■ Appendix A, "System Log and Configuration Files"

Other administration topics that may be useful to Oracle Real-Time Decisions administrators are covered in other Oracle manuals. For more information, see Section 1.3, "Topics of Interest in Other Guides."

1.2.2 Oracle Real-Time Decisions as an Oracle Business Intelligence Component

Figure 1–3 illustrates a typical configuration of the Oracle Business Intelligence components, which include Oracle Real-Time Decisions, that share a common administration framework. Oracle Real-Time Decisions *(shown and highlighted in the diagram as RTD)* is an Oracle Fusion Middleware Java component which is deployed as one or more Java EE applications and a set of resources.

Figure 1–3 Oracle Business Intelligence Components


Oracle BI Domain - Enterprise Installation
Java Components (WebLogic Domain)
Administration Server
WLS Administration Console
Fusion Middleware Control
JMX MBeans
Managed Server
Action Services
EAS
Workspace
BI Publisher
BI Composer
BI Office
Financial Reporting
Web Service SOA
Security Services
Calc Manager
RTD
Hyperion Provider Services
BI Presentation Services Plug-in
Node Manager
Oracle BI System Components (BI Instance)
BI Server
BI Scheduler
BI Presentation Services
BI JavaHost
Cluster Controller
Essbase Studio Server
Essbase Agent
Essbase App
OPMN
Configuration Files, BI Repository, Essbase Repository, Presentation Catalog, Security, Identity Management, Connection Configuration, Database Configuration, Log Files

For more information, see Section 1.2.3, "Oracle Real-Time Decisions Configured in a WebLogic Domain" and *Oracle Fusion Middleware Administrator's Guide*.

1.2.3 Oracle Real-Time Decisions Configured in a WebLogic Domain

Oracle Real-Time Decisions is configured in a WebLogic domain which contains the Java components required to run and administer Oracle Real-Time Decisions.

The rest of this section introduces and includes brief descriptions of the main components of WebLogic domains. For more detailed descriptions, see the chapter "Understanding Oracle Fusion Middleware Concepts" in *Oracle Fusion Middleware Administrator's Guide*.

Administration Server

The Administration Server is a standard JMX MBean container that provides local and centralized management of the domain where Oracle RTD is deployed.

The Administration Server is hosted in the Middleware Home which is part of the WebLogic domain, and contains the Oracle WebLogic Server Administration Console.

The Administration Server can access functions contained within Managed Servers on remote physical machines in order to achieve centralized configuration and provisioning of all the components within the domain where Oracle RTD is deployed.

The main tools to interact with the Administration Server are the following:

- Oracle WebLogic Server Administration Console
- Oracle WebLogic Scripting Tool (WLST)

For more information, see Section 1.2.4, "Which System Administration Tools Enable You to Manage Oracle RTD?".

Managed Server

A Managed Server is an individual JEE application container hosted within each WebLogic instance. It provides local management functions on individual machines for Java and system components contained within the local Middleware home. It refers to the Administration Server for all of its configuration and deployment information.

Node Manager

Oracle WebLogic Server includes Node Manager, a daemon process that provides remote server start, stop and restart capabilities when processes become unresponsive or terminate unexpectedly.

For more information, see *Oracle Fusion Middleware Node Manager Administrator's Guide for Oracle WebLogic Server*.

> **Note:** Figure 1–3 shows a typical component configuration after an Enterprise installation of Oracle Business Intelligence, with Oracle RTD configured in a Managed Server.
>
> Another component configuration is where all the individual Java components, apart from the Node Manager, are contained in the Administration Server - as occurs after a Simple installation of Oracle Business Intelligence. For more information, see *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

1.2.4 Which System Administration Tools Enable You to Manage Oracle RTD?

The following system administration tools enable you to manage Oracle RTD:

- Section 1.2.4.1, "Oracle Enterprise Manager Fusion Middleware Control"
- Section 1.2.4.2, "Oracle WebLogic Server Administration Console"
- Section 1.2.4.3, "Oracle WebLogic Scripting Tool (WLST)"

1.2.4.1 Oracle Enterprise Manager Fusion Middleware Control

Oracle Enterprise Manager Fusion Middleware Control is a browser-based tool that enables you to monitor and configure Oracle Fusion Middleware components. You can deploy applications, manage security, and create Oracle Fusion Middleware clusters. For more information, see Section 2.1, "Using Fusion Middleware Control to Manage Oracle Real-Time Decisions."

1.2.4.2 Oracle WebLogic Server Administration Console

Oracle WebLogic Server is a Java EE application server that supports the deployment of JEE Java components in a secure, highly available, and scalable environment.

You use the Oracle WebLogic Server Administration Console to manage and monitor a WebLogic Server domain. For more information, see Section 2.2, "Using Oracle WebLogic Server Administration Console to Manage Oracle RTD."

For more information about Oracle WebLogic Server, see Oracle Technology Network on `http://www.oracle.com/technology/index.html`.

1.2.4.3 Oracle WebLogic Scripting Tool (WLST)

The Oracle WebLogic Scripting Tool (WLST) is a command-line scripting environment that you can use to administer Oracle Real-Time Decisions. The WLST scripting environment is based on the Java scripting interpreter Jython. You can use this tool interactively on the command line; in batch scripts supplied in a file (Script Mode, where scripts invoke a sequence of WLST commands without requiring your input), or embedded in Java code. You can extend the WebLogic scripting language by following the Jython language syntax.

For more information, see *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

1.3 Topics of Interest in Other Guides

Some topics that may be of interest to system administrators are covered in other guides. Table 1–2 lists these topics, and indicates where to go for more information.

Table 1–2 Topics Covered in Other Guides

Topic	Where To Go For More Information
Installation	*Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*
Moving from a test to a production system	*Oracle Fusion Middleware Administrator's Guide*
General administration under Oracle Fusion Middleware	*Oracle Fusion Middleware Administrator's Guide*
Upgrading from Oracle RTD Version 3.0.0.1	Appendix B, "Upgrading and Patching Oracle Real-Time Decisions" and *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*
High availability	*Oracle Fusion Middleware High Availability Guide*
Backup and recovery	*Oracle Fusion Middleware Administrator's Guide*
Third-party tools and relational data source adapters	Section 1.4, "System Requirements and Certification"

1.4 System Requirements and Certification

Refer to the system requirements and certification documentation for information about hardware and software requirements, platforms, databases, and other information. Both of these documents are available on Oracle Technology Network (OTN).

The system requirements document covers information such as hardware and software requirements, minimum disk space and memory requirements, and required system libraries, packages, or patches:

`http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html`

The certification document covers supported installation types, platforms, operating systems, databases, JDKs, and third-party products:

`http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html`

2 Main Administration Tools for Oracle Real-Time Decisions

This chapter introduces and summarizes how to manage Oracle RTD using the two main administration tools for Oracle Real-Time Decisions: Fusion Middleware Control and Oracle WebLogic Server Administration Console.

This chapter includes the following topics:

- Section 2.1, "Using Fusion Middleware Control to Manage Oracle Real-Time Decisions"
- Section 2.2, "Using Oracle WebLogic Server Administration Console to Manage Oracle RTD"
- Section 2.3, "General Administration Tasks"

2.1 Using Fusion Middleware Control to Manage Oracle Real-Time Decisions

This section contains the following topics:

- Section 2.1.1, "Logging into Fusion Middleware Control"
- Section 2.1.2, "Administering Oracle RTD Using Fusion Middleware Control"

2.1.1 Logging into Fusion Middleware Control

To log into Fusion Middleware Control, you enter the Fusion Middleware Control URL, in the format:

`http://`*`hostname`*`:`*`port`*`/em`

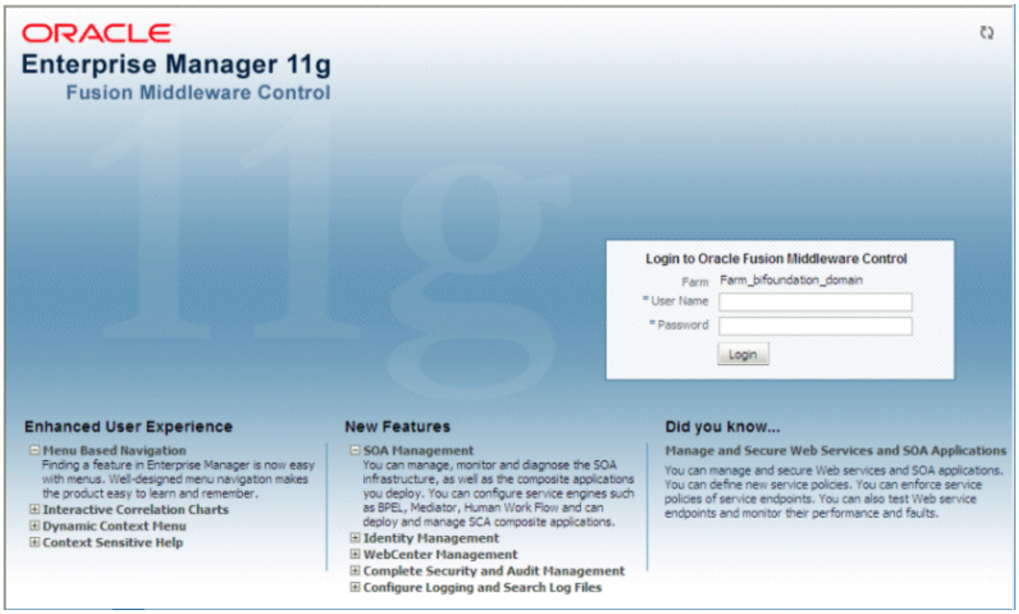
The port number is the number of the Administration Server. By default, the port number is 7001.

To login to Fusion Middleware Control:

1. Enter the URL in your Web browser. For example:

   ```
   http://host1.example.com:7001/em
   ```

   The Fusion Middleware Control login page is displayed.


ORACLE
Enterprise Manager 11g
Fusion Middleware Control
Login to Oracle Fusion Middleware Control
Login
Enhanced User Experience
⊟ Menu Based Navigation
⊞ Interactive Correlation Charts
⊞ Dynamic Context Menu
⊞ Context Sensitive Help
New Features
⊟ SOA Management
⊞ Identity Management
⊞ WebCenter Management
⊞ Complete Security and Audit Management
⊞ Configure Logging and Search Log Files
Did you know...
Manage and Secure Web Services and SOA Applications

Farm: Farm_bifoundation_domain

* User Name:

* Password:

Finding a feature in Enterprise Manager is now easy with menus. Well-designed menu navigation makes the product easy to learn and remember.

You can manage, monitor and diagnose the SOA infrastructure, as well as the composite applications you deploy. You can configure service engines such as BPEL, Mediator, Human Work Flow and can deploy and manage SCA composite applications.

You can manage and secure Web services and SOA applications. You can define new service policies. You can enforce service policies of service endpoints. You can also test Web service endpoints and monitor their performance and faults.

2. Enter the Oracle Fusion Middleware administrator user name and password and click **Login**.

> **Note:** *The folders available in the Fusion Middleware Control left-hand side Target Navigation Pane vary depending on what products and options were chosen during installation. The descriptions in this section assume the presence of the folders that are available after a simple "default" installation, where Oracle RTD is installed into a cluster called bi_cluster.*

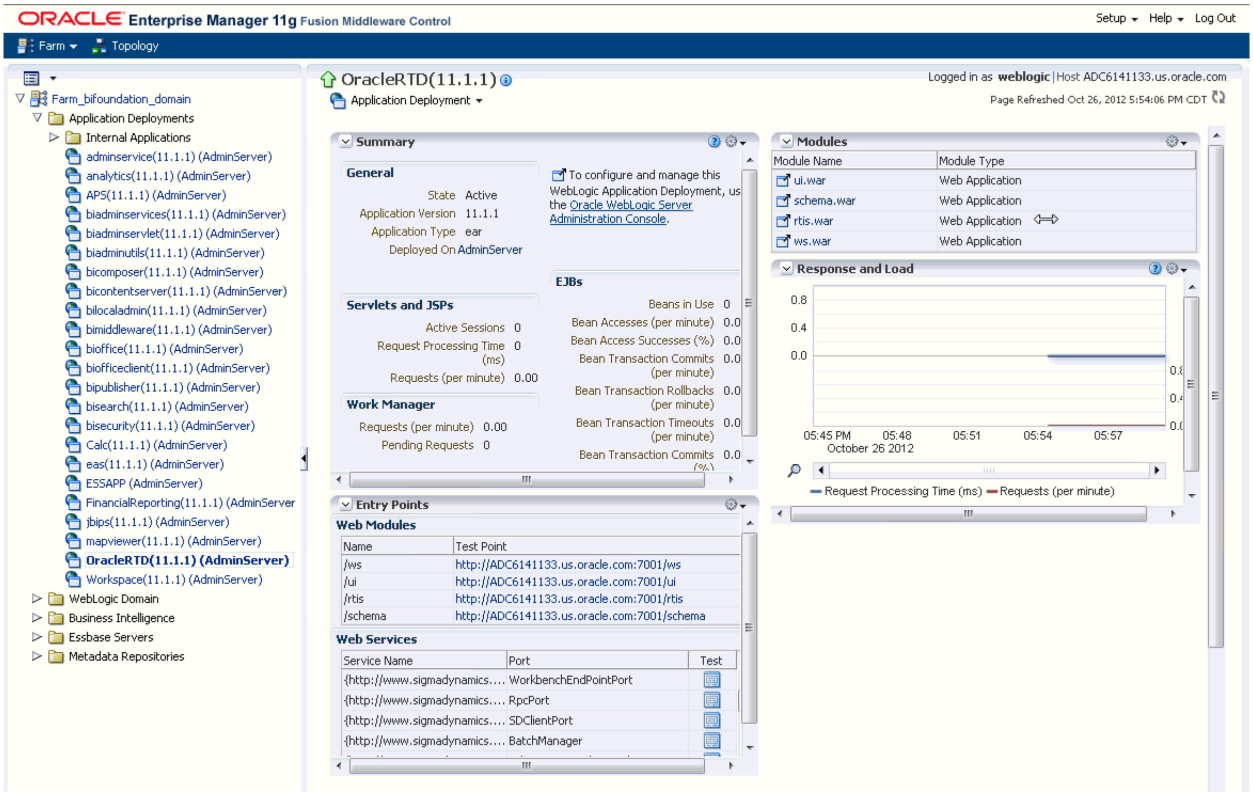
Accessing the Oracle Real-Time Decisions Home Page

The Oracle Real-Time Decisions home page shows the Oracle RTD components and general system status information.

You can access the Oracle RTD home page through several paths, the main one being the following:

- Expand the **Application Developments** folder and select the Oracle RTD node (as shown in Figure 2–1) or the Oracle RTD node within the cluster or server that contains Oracle RTD.

Figure 2–1 Oracle RTD Home Page in Fusion Middleware Control



ORACLE Enterprise Manager 11g Fusion Middleware Control

Setup ▾ Help ▾ Log Out

Farm ▾ Topology

- Farm_bifoundation_domain
  - Application Deployments
    - Internal Applications
    - adminservice(11.1.1) (AdminServer)
    - analytics(11.1.1) (AdminServer)
    - APS(11.1.1) (AdminServer)
    - biadminservices(11.1.1) (AdminServer)
    - biadminservlet(11.1.1) (AdminServer)
    - biadminutils(11.1.1) (AdminServer)
    - bicomposer(11.1.1) (AdminServer)
    - bicontentserver(11.1.1) (AdminServer)
    - bilocaladmin(11.1.1) (AdminServer)
    - bimiddleware(11.1.1) (AdminServer)
    - bioffice(11.1.1) (AdminServer)
    - biofficeclient(11.1.1) (AdminServer)
    - bipublisher(11.1.1) (AdminServer)
    - bisearch(11.1.1) (AdminServer)
    - bisecurity(11.1.1) (AdminServer)
    - Calc(11.1.1) (AdminServer)
    - eas(11.1.1) (AdminServer)
    - ESSAPP (AdminServer)
    - FinancialReporting(11.1.1) (AdminServer
    - jbips(11.1.1) (AdminServer)
    - mapviewer(11.1.1) (AdminServer)
    - **OracleRTD(11.1.1) (AdminServer)**
    - Workspace(11.1.1) (AdminServer)
  - WebLogic Domain
  - Business Intelligence
  - Essbase Servers
  - Metadata Repositories

OracleRTD(11.1.1)

Application Deployment ▾

Logged in as **weblogic** | Host ADC6141133.us.oracle.com

Page Refreshed Oct 26, 2012 5:54:06 PM CDT

Summary

General

State Active
Application Version 11.1.1
Application Type ear
Deployed On AdminServer

To configure and manage this WebLogic Application Deployment, us the Oracle WebLogic Server Administration Console.

Servlets and JSPs

Active Sessions 0
Request Processing Time (ms) 0
Requests (per minute) 0.00

Work Manager

Requests (per minute) 0.00
Pending Requests 0

EJBs

Beans in Use 0
Bean Accesses (per minute) 0.0
Bean Access Successes (%) 0.0
Bean Transaction Commits (per minute) 0.0
Bean Transaction Rollbacks (per minute) 0.0
Bean Transaction Timeouts (per minute) 0.0
Bean Transaction Commits (%) 0.0

Modules

Module Name	Module Type
ui.war	Web Application
schema.war	Web Application
rtis.war	Web Application
ws.war	Web Application

Response and Load

Entry Points

Web Modules

Name	Test Point
/ws	http://ADC6141133.us.oracle.com:7001/ws
/ui	http://ADC6141133.us.oracle.com:7001/ui
/rtis	http://ADC6141133.us.oracle.com:7001/rtis
/schema	http://ADC6141133.us.oracle.com:7001/schema

Web Services

Service Name	Port	Test
{http://www.sigmadynamics....	WorkbenchEndPointPort	
{http://www.sigmadynamics....	RpcPort	
{http://www.sigmadynamics....	SDClientPort	
{http://www.sigmadynamics....	BatchManager	

You can also access the Oracle RTD home page through the **WebLogic Domain** folder. Select the domain node (default value is bifoundation_domain), then select the Oracle RTD node in the list of deployments for the cluster or server.

2.1.2 Administering Oracle RTD Using Fusion Middleware Control

While general component and status information displayed on the Oracle Real-Time Decisions home page, Oracle Real-Time Decisions administrative task options are available from an untitled pane of options, accessible through either of the following methods:

- From the Fusion Middleware Control Target Navigation Pane, right-click the Oracle RTD node wherever it appears under the expanded **Application Developments** or **WebLogic Domain** folders
- From the Oracle RTD home page, select the **Application Development** subheading directly under the Oracle RTD heading

Table 2–1 shows the main administrative task options available from this pane.

Table 2–1 Main Oracle RTD Administration Task Options in Fusion Middleware Control

Option	Tasks	More Information
Control	Starting and stopping Oracle Real-Time Decisions	■ Section 2.3.2, "Starting and Stopping Oracle RTD"
Logs	Viewing Oracle Real-Time Decisions log messages and configuring Oracle Real-Time Decisions log files	■ Appendix A, "System Log and Configuration Files"

Table 2–1 (Cont.) Main Oracle RTD Administration Task Options in Fusion Middleware

Option	Tasks	More Information
Performance Summary	Viewing Oracle Real-Time Decisions performance statistics	■ Chapter 11, "Performance Monitoring"
Web Services	Viewing and editing Web Service details	■ Appendix "Oracle Real-Time Decisions Web Services and Clients" in *Oracle Fusion Middleware Platform Developer's Guide for Oracle Real-Time Decisions*
Security	Managing application policies and application roles for Oracle RTD	■ Chapter 4, "Security for Oracle Real-Time Decisions"
System MBean Browser	Managing Oracle RTD through MBeans	■ Chapter 13, "Managing Oracle Real-Time Decisions"

See *Oracle Fusion Middleware Administrator's Guide* for additional information about how to use Fusion Middleware Control.

2.2 Using Oracle WebLogic Server Administration Console to Manage Oracle RTD

You use the Oracle WebLogic Server Administration Console to administer general components that affect Oracle RTD, such as users, groups, data sources, and clusters.

You display Oracle WebLogic Server Administration Console, using one of the following methods:

- Using the Start menu in Windows
- Clicking a link on the Overview page in Fusion Middleware Control
- Entering a URL into a Web browser window

To login to Oracle WebLogic Server Administration Console:

1. If the Administration Server for WebLogic Server is not running, start it.

   For more information, see Section 2.3.1, "Starting and Stopping Oracle WebLogic Server Instances."

2. Display the Oracle WebLogic Server Administration Console using one of the following methods:

   Clicking a link on the Overview page in Fusion Middleware Control:

   a. Log into Fusion Middleware Control (for more information, see Section 2.1.1, "Logging into Fusion Middleware Control").

   b. Expand the WebLogic Domain node and select the domain that contains Oracle RTD.

   c. Click the link entitled **Oracle WebLogic Server Administration Console** in the Summary region.

      The Oracle WebLogic Server Administration Console login page is displayed.

   Using a URL in a Web browser window:

   a. Start a Web browser.

   b. Enter the following URL into the browser:

`http://`*`hostname`*`:`*`port`*`/console`

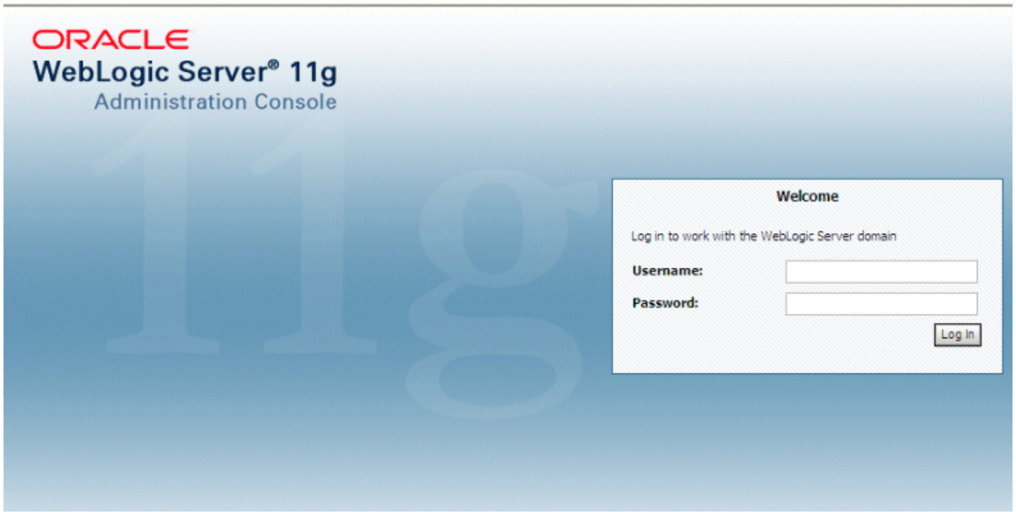
For example, `http://mymachine:7001/console/`.

where `hostname` is the DNS name or IP address of the Administration Server and `port` is the port on which the Administration Server is listening for requests (port 7001 by default). If you have configured a domain-wide Administration port, then use that port number. If you configured the Administration Server to use Secure Socket Layer (SSL), then you must add s after http as follows:

`https://`*`hostname`*`:`*`port`*`/console`

Note: A domain-wide administration port always uses SSL.

The Oracle WebLogic Server Administration Console login page is displayed.

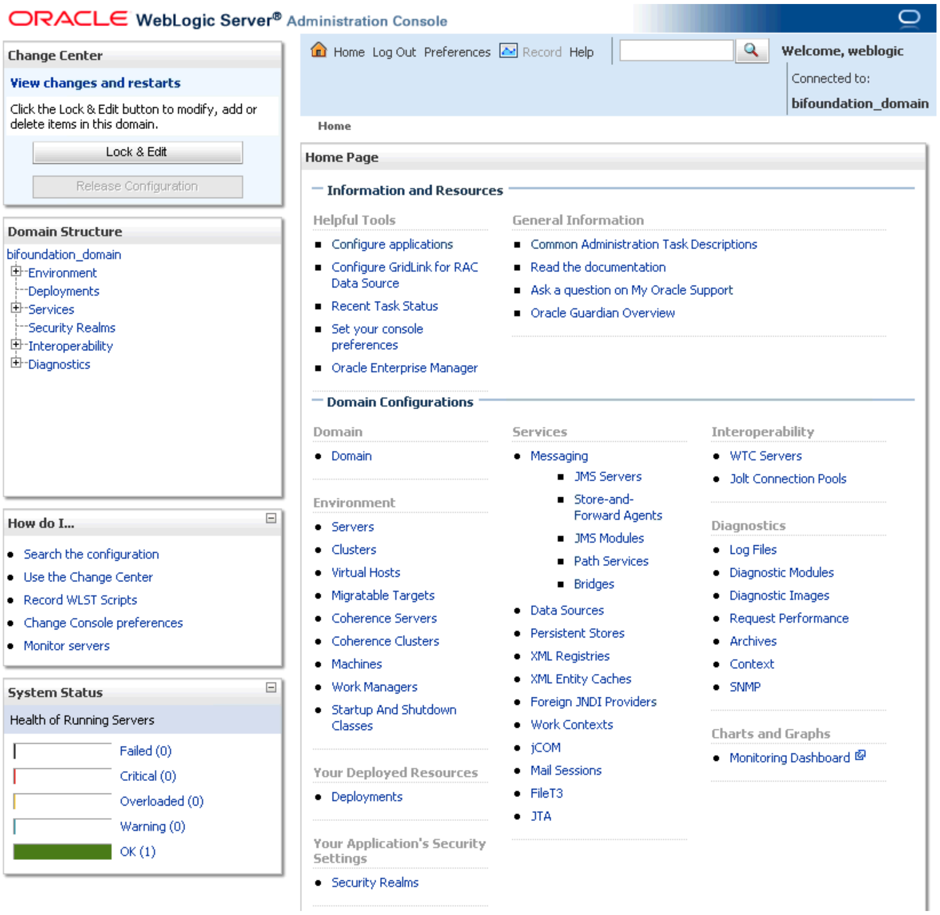


3. Enter the system administrator username and password and click Login.

   This system-wide administration user name and password was specified during the installation process, and can be used to login to WebLogic Server Administration Console and Fusion Middleware Control. Alternatively enter a user name that belongs to the Administrators or BIAdministrators security group.

   Note: If you have your browser configured to send HTTP requests to a proxy server, then you might need to configure your browser to not send Oracle WebLogic Server Administration Server HTTP requests to the proxy. If the Oracle WebLogic Server Administration Server is on the same machine as the browser, then ensure that requests sent to localhost or 127.0.0.1 are not sent to the proxy.

   Oracle WebLogic Server Administration Console displays the Home page, connected to the bifoundation_domain.



Click the Lock & Edit button to modify, add or delete items in this domain.

For more information on Oracle WebLogic administration, see the following:

- *Oracle Fusion Middleware Information Roadmap for Oracle WebLogic Server*
- *Oracle Fusion Middleware Introduction to Oracle WebLogic Server*
- *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*

2.3 General Administration Tasks

The Administration Server for Oracle WebLogic Server and any Managed Server that contains Oracle Real-Time Decisions must be running before you can start Oracle Real-Time Decisions.

This section explains how to start and stop the Administration Server and Managed Servers for Oracle WebLogic Server, and how to start and stop Oracle Real-Time Decisions.

This section contains the following topics:

- Section 2.3.1, "Starting and Stopping Oracle WebLogic Server Instances"
- Section 2.3.2, "Starting and Stopping Oracle RTD"

2.3.1 Starting and Stopping Oracle WebLogic Server Instances

Oracle WebLogic Server provides several ways to start and stop the Administration Server and Managed Server instances. This section describes how to:

- Start and stop the Administration Server with scripts
- Start and stop Managed Servers using the Oracle WebLogic Server Administration Console

For more information, see the *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

2.3.1.1 Starting and Stopping the Administration Server

This topic explains how to start the Administration Server for Oracle WebLogic Server.

Note: When you start the Administration Server, the database you specified during install must be running, or JDBC errors will prevent startup.

Use the method appropriate for your operating system:

- On Windows, use one of the following methods:
  - In the Windows Start Menu, go to All Programs > Oracle WebLogic > User Projects > bifoundation_domain >

    Start Admin Server for WebLogic Server Domain

    Stop Admin Server for WebLogic Server Domain

    In Windows, Oracle displays an MSDOS progress window that indicates the progress of the processing steps of starting the Administration Server.
  - Open a DOS prompt and change the directory to `<`*mw_home*`>\user_projects\domains\bifoundation_domain\bin`. Then, run one of the following commands to start or stop:

    ```
    startWebLogic.cmd

    stopWebLogic.cmd
    ```

    Note: You can also stop the Java components in the MS-DOS window where the Java process was started, if you press the Ctrl+C key combination
- On Linux or UNIX, open a shell prompt and change the directory to `<`*mw_home*`>/user_projects/domains/bifoundation_domain/bin`. Then, run one of the following commands to start or stop:

  ```
  ./startWebLogic.sh

  ./stopWebLogic.sh
  ```

  Note: On Linux or UNIX, you can also use the process termination command for the operating system in use (for example, kill on Linux or UNIX). Java indicates on the console window that it is shutting down when it receives a shutdown signal.

Note: When you start the Administration Server, you should confirm that it is running, by trying to log into the Oracle WebLogic Server Administration Console. For more information, see Section 2.2, "Using Oracle WebLogic Server Administration Console to Manage Oracle RTD."

2.3.1.2 Starting and Stopping Managed Servers

1. Log into the Oracle WebLogic Server Administration Console.

   For more information, see Section 2.2, "Using Oracle WebLogic Server Administration Console to Manage Oracle RTD."
2. In the Domain Structure region, click Environment, then Servers.

The Oracle WebLogic Server Administration Console displays the Summary of Servers page.

3. Select the Control tab, then select the check box beside the Managed Server that you want to start or stop.
4. Click Start or Stop to start or stop the Managed Server as required.

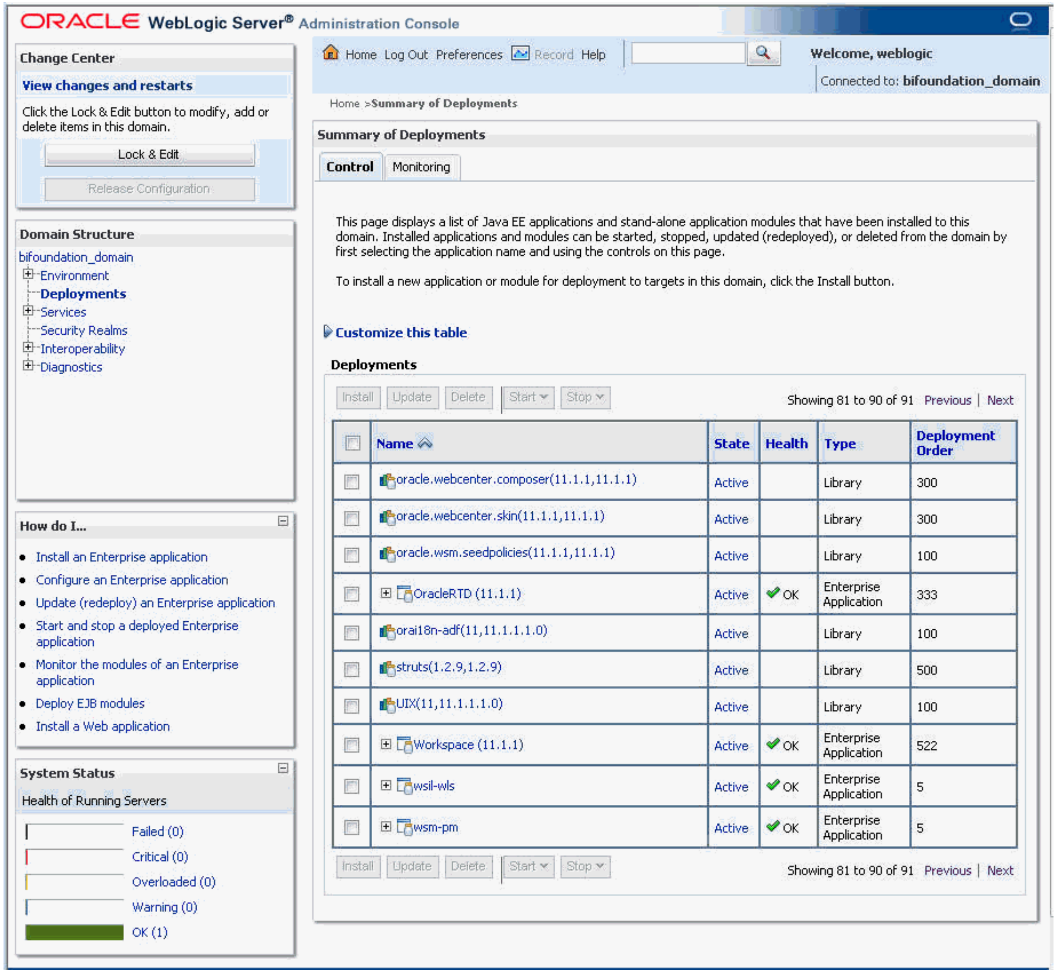
2.3.2 Starting and Stopping Oracle RTD

To start and stop Oracle Real-Time Decisions:

1. Log into the Oracle WebLogic Server Administration Console.

   For more information, see Section 2.2, "Using Oracle WebLogic Server Administration Console to Manage Oracle RTD."
2. In the Domain Structure region, click Deployments.

   The Oracle WebLogic Server Administration Console displays the Summary of Deployments page.
3. Locate the page that contains Oracle RTD.
4. Display the Control tab.
5. Select the check box beside Oracle RTD.



Click the Lock & Edit button to modify, add or delete items in this domain.

- Install an Enterprise application
- Configure an Enterprise application
- Update (redeploy) an Enterprise application
- Start and stop a deployed Enterprise application
- Monitor the modules of an Enterprise application
- Deploy EJB modules
- Install a Web application

This page displays a list of Java EE applications and stand-alone application modules that have been installed to this domain. Installed applications and modules can be started, stopped, updated (redeployed), or deleted from the domain by first selecting the application name and using the controls on this page.

To install a new application or module for deployment to targets in this domain, click the Install button.

□	Name	State	Health	Type	Deployment Order
□	oracle.webcenter.composer(11.1.1,11.1.1)	Active		Library	300
□	oracle.webcenter.skin(11.1.1,11.1.1)	Active		Library	300
□	oracle.wsm.seedpolicies(11.1.1,11.1.1)	Active		Library	100
□	⊞ OracleRTD (11.1.1)	Active	✓ OK	Enterprise Application	333
□	orai18n-adf(11,11.1.1.1.0)	Active		Library	100
□	struts(1.2.9,1.2.9)	Active		Library	500
□	UIX(11,11.1.1.1.0)	Active		Library	100
□	⊞ Workspace (11.1.1)	Active	✓ OK	Enterprise Application	522
□	⊞ wsil-wls	Active	✓ OK	Enterprise Application	5
□	⊞ wsm-pm	Active	✓ OK	Enterprise Application	5

6. Click Start or Stop to start or stop Oracle RTD as required.

3 Post-Installation Steps

This chapter describes the file and directory structure that exists after a Simple or Enterprise install of Oracle RTD, and the configuration steps that may be performed after a Software Only install.

Installation of Oracle RTD is performed using the Oracle Fusion Middleware Business Intelligence Installer.

There are three types of installation:

- Simple Install
- Enterprise Install
- Software Only Install

In each of the install scenarios in the Business Intelligence Installer, you have the option of selecting different Business Intelligence components. The selection and installation of Oracle RTD is independent of the selection and installation of any of the other components.

The Simple and Enterprise installs are similar in that the same types of system object are created and configured. The Simple install uses default values, the Enterprise install enables installers to select many non-default values. The Software Only install copies component files into standard directories, but neither configures nor deploys any component.

Full details of how to install Oracle RTD appear in *Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*.

The CrossSell and DC_Demo Inline Services are released with Oracle RTD to serve as examples to demonstrate a variety of Oracle RTD features. The data required for these Inline Services is provided with Oracle RTD, but not set up during installation. This chapter describes how to set up the data for these sample Inline Services.

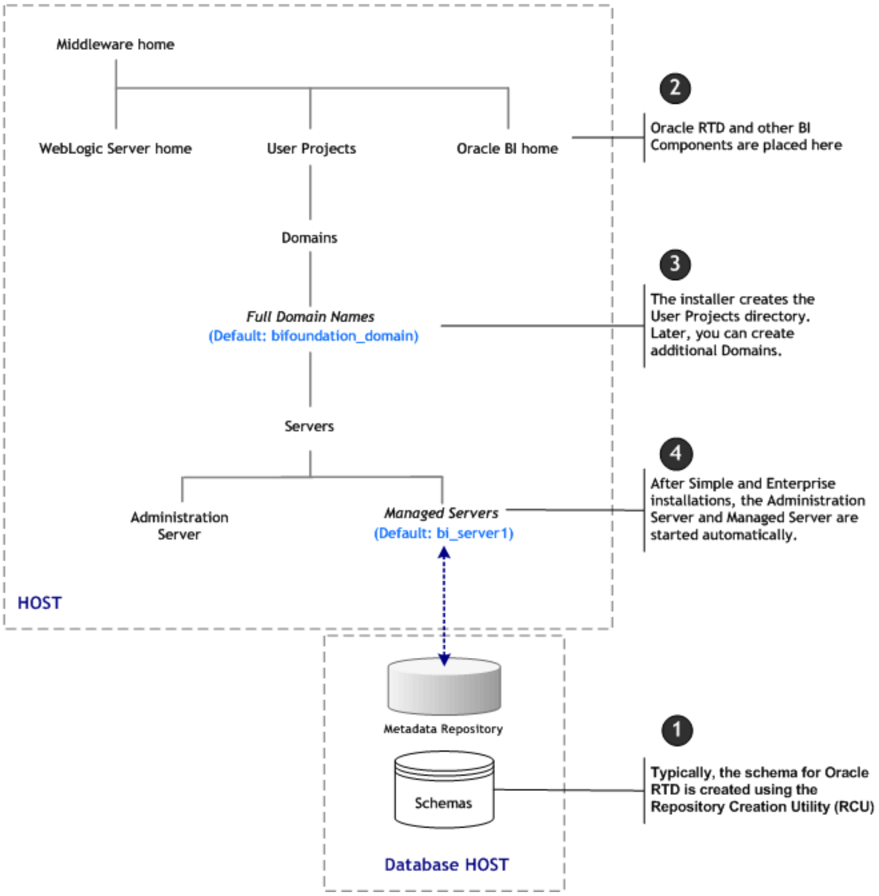
This chapter contains the following topics:

3.1 Directory Structure of Oracle RTD Server-Side Files

Figure 3–1 shows the main server-side directories after a basic Simple or Enterprise installation, no matter which Business Intelligence product was installed.

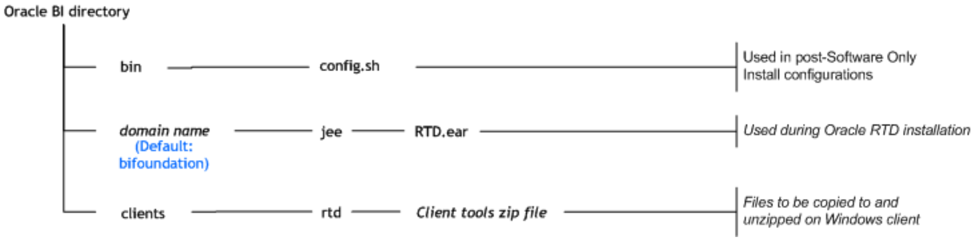
> **Note:** The exact names of the directories and files may depend on options chosen during installation. Unless specified otherwise, the diagrams and descriptions in this chapter use the standard default names created during a Simple installation.

Figure 3–1 Main Server-Side Directories


Middleware home
WebLogic Server home
User Projects
Oracle BI home
Domains
Full Domain Names
(Default: bifoundation_domain)
Servers
Administration Server
Managed Servers
(Default: bi_server1)
HOST
Metadata Repository
Schemas
Database HOST
2
Oracle RTD and other BI Components are placed here
3
The installer creates the User Projects directory. Later, you can create additional Domains.
4
After Simple and Enterprise installations, the Administration Server and Managed Server are started automatically.
1
Typically, the schema for Oracle RTD is created using the Repository Creation Utility (RCU)

The Oracle BI directory is the home directory for the installed products. In this section, it is referred to as *<Oracle_BI_directory>*, and contains both product-specific files and files common to all installed products. Figure 3–2 shows the subdirectories and files under *<Oracle_BI_directory>* that are most relevant to Oracle RTD administrators.

Figure 3–2 Oracle BI Directory - Main Subdirectories


Oracle BI directory
bin
config.sh
Used in post-Software Only Install configurations
domain name (Default: bifoundation)
jee
RTD.ear
Used during Oracle RTD installation
clients
rtd
Client tools zip file
Files to be copied to and unzipped on Windows client

After the completion of both the Simple and Enterprise installations:

- The `RTD.ear` file will be at *`<Oracle_BI_directory>`*`/bifoundation/jee`.
- The system will consist of a new WebLogic domain with one Administration Server and one Managed Server.
- Oracle RTD will be configured and deployed in the Managed Server.
- The Oracle RTD client-side tools will exist in a zip file in the directory *`<Oracle_BI_directory>`*`/clients/rtd/`.

  The name of the client tools zip file is `rtd_client_11.1.1.zip`.

  The Oracle RTD client-side tools can only be run on a Windows platform. If you did not install Oracle RTD on to a Windows platform, you must copy the client tools zip file to a Windows client machine, and unzip it there. For more information, see Section 3.2, "Installing Oracle RTD Client-Side Files."

The tables and procedures required to run Oracle RTD are typically created previously by running Repository Creation Utility (RCU). The tables include the model snapshot tables, which reside in the same schema as the other Oracle RTD tables.

Notes: **1.** While RCU creates the model snapshot tables in the same schema as the Oracle RTD runtime tables, Oracle recommends that you also create and configure the model snapshot tables in a separate schema. For details, see Chapter 10, "Setting Up and Using Model Snapshots."

2. The example Inline Services CrossSell and DC_Demo, which are included with Oracle RTD, refer to specific example data tables. Before you can use these Inline Services, you must create and populate the example data tables. For details of creating and populating these tables, see Section 3.5.1, "Populating the CrossSell Example Data (Optional)" and Section 3.5.2, "Populating the DC_Demo Example Data (Optional).".

3.2 Installing Oracle RTD Client-Side Files

Oracle RTD can run on either Windows or UNIX. The Oracle RTD client tools, such as Decision Studio and Load Generator can only run on Windows. The process of installing the Oracle RTD client tools is unzipping the client tools zip file on a Windows machine - no further procedures are required.

If you installed Oracle RTD on to a Windows platform, unzip the client tools zip file located in the server-side directory *`<Oracle_BI_directory>`*`/clients/rtd/` into any directory that you choose.

If you did not install Oracle RTD on to a Windows platform, you must perform the two-stage process:

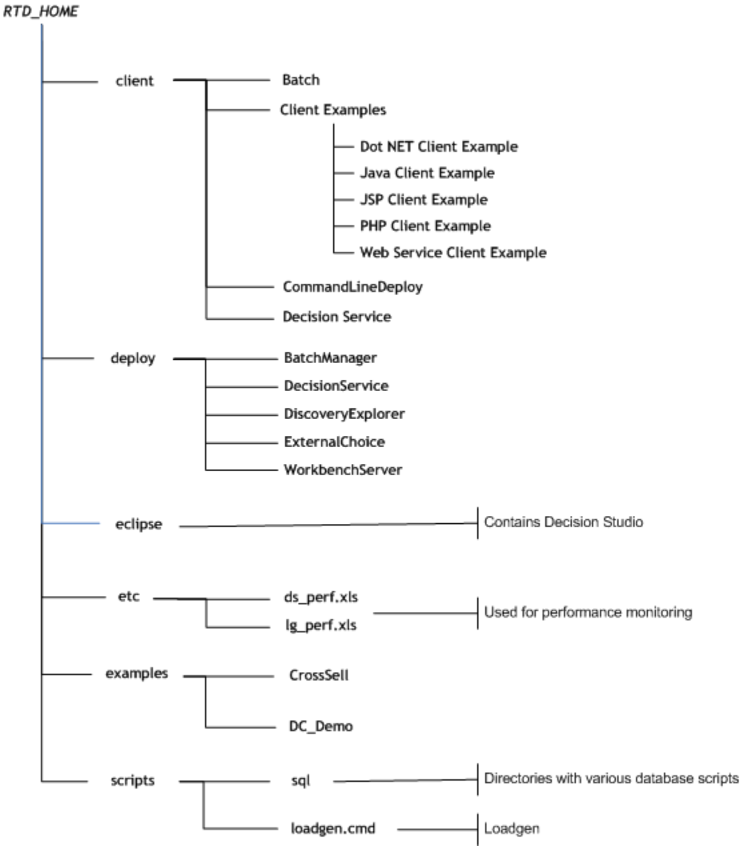
- Copy the client tools zip file from the server-side directory *<Oracle_BI_directory>*`/clients/rtd/` to the Windows client machine where you want to use the client tools
- Unzip the client tools zip file on the Windows client machine into any directory that you choose

For more information about supported machine configurations for Oracle RTD and Oracle RTD client tools, see Section 1.4, "System Requirements and Certification."

> **Note:** The terminology convention used for all Oracle RTD documentation is that the directory into which the Oracle RTD client-side tools are installed is referred to as ***RTD_HOME***. See also Section 3.4, "About the Oracle RTD Runtime Environment."

Figure 3–3 shows the main Oracle RTD client-side directories and files after the client tools zip file has been unzipped.

Figure 3–3 Main Oracle RTD Client Directories and Files



3.2.1 Installing Java Development Kit (JDK) for Oracle RTD Client Tools

To use the Oracle RTD client-side tools after installation, ensure that you have the same version of the Java Development Kit (JDK) as you have on the server machine where you installed Oracle RTD. The server-side JDK typically resides in a directory directly under the *<middleware_home>* directory.

After installing the JDK on the client machine, create a system environment variable called `JAVA_HOME` and set its value to the full path name of the install location of the JDK.

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

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

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

By default, setting the `JAVA_HOME` system environment variable will enable all the Oracle RTD client tools. You can also set `JAVA_HOME` locally, in the file *RTD_HOME*`/scripts/SetSDParams.cmd.` This local setting of `JAVA_HOME` only affects Oracle RTD operations that use `sdexec.cmd`, such as `InitAppdb.cmd` and `loadgen.cmd` (and not Decision Studio nor the command line deployer).

3.3 Configuring Oracle RTD After Installation

Configuring Oracle RTD After Software Only Installs

The Software Only install installs the binaries of components selected during the installation into the appropriate directories. It makes some minor changes to directories under Middleware home, but the main effect is to create a new Oracle BI directory, and to set up product-related files under that new directory.

The Software Only install performs no configuration nor enabling of any component. No associated WebLogic domain is created. JEE components, such as Oracle RTD, are neither configured nor deployed.

To complete the process of configuring Oracle RTD, and creating a WebLogic domain and all the default security components, you must run *<Oracle_BI_directory>*`/bin/config.sh` (on Linux systems) or *<Oracle_BI_directory>*`/bin/config.bat` (on Windows systems).

Configuring Oracle RTD After Simple and Enterprise Installs

A Simple or Enterprise installation automatically creates a single WebLogic domain for Oracle RTD (and any other components selected during the install). If you do not need further WebLogic domains for Oracle RTD, you do not have to perform any post-installation steps.

After a Simple or Enterprise installation, if you require more WebLogic domains for Oracle RTD under the same *<Oracle_BI_directory>*, run the script *<Oracle_BI_directory>*`/bin/config.sh` (on Linux systems) or *<Oracle_BI_directory>*`/bin/config.bat` (on Windows systems).

3.4 About the Oracle RTD Runtime Environment

The files that are used and updated during Oracle RTD runtime appear at *<mw_home>*`/user_projects/domains/`*domain_name*`/servers/`*server_name*`/`, and for convenience, this directory is referred to as *RTD_RUNTIME_HOME*.

As an example, the Oracle RTD logs and other Managed Server logs appear in *RTD_RUNTIME_HOME*`/logs/`.

3.5 Populating the Example Data Tables (Optional)

Two example Inline Services, CrossSell and DC_Demo, are included with Oracle Real-Time Decisions. To use either or both of these Inline Services, you must create and populate extra tables in the database, by running the script `InitAppDB` on the Windows computer where you installed the Oracle RTD client-side tools.

Database Client Tools Requirement

On the Windows computer, in addition to the Oracle RTD client-side tools, you must also have client tools appropriate to your database. This can be achieved by installing:

- SQL Server client tools if you are using SQL Server for your Oracle RTD Database
- Oracle Administrator Client (must include SQL*Plus) if you are using Oracle Database for your Oracle RTD Database
- DB2 Administration Client if you are using DB2 for your Oracle RTD Database

This section contains the following topics:

- Section 3.5.1, "Populating the CrossSell Example Data (Optional)"
- Section 3.5.2, "Populating the DC_Demo Example Data (Optional)"

3.5.1 Populating the CrossSell Example Data (Optional)

An example Inline Service, called CrossSell, is included with Oracle Real-Time Decisions. To use this sample Inline Service, you must create and populate three tables, `CrossSellCustomers`, `CrossSellResponses`, and `CrossSellBestOffer` in the Oracle RTD Database.

`InitAppDB` is located with the example Inline Service. Using a command prompt, run the script appropriate for your database type:

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

This script takes the following parameters:

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

Table 3–1 describes the parameters for the `InitAppDB` script.

Table 3–1 Parameters for InitAppDB Script

Parameter	Description
RTD_HOME	The full path of the directory where the Oracle RTD client-side files are installed.
db_host	The name of the computer hosting the database server. If you installed your Oracle RTD Database on a SQL Server named instance, specify *db_host*`\`*instance_name*.
db_port	The database port number.
db_name	The name of the database, or for Oracle Database, the SID.

Table 3–1 (Cont.) Parameters for InitAppDB Script

Parameter	Description
db_runtime_user[1]	The user name of the run-time user for the system.
db_admin_user	The name of a user that has rights to create tables and stored procedures on the database.
db_admin_password	The password of the administrative user.

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

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

3.5.2 Populating the DC_Demo Example Data (Optional)

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

`InitAppDB` is located with the example Inline Service. Using a command prompt, run the script appropriate for your database type:

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

This script takes the following parameters:

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

See Table 3–1 for the parameters for the `InitAppDB` script.

4

Security for Oracle Real-Time Decisions

Oracle Real-Time Decisions integrates seamlessly with the Oracle Fusion Middleware platform and they share a common security framework and features. This chapter includes an overview of the security framework to provide background for understanding the overall security model.

For more information about the Oracle Fusion Middleware platform and the common security framework, see *Oracle Fusion Middleware Application Security Guide*.

This chapter contains the following sections:

4.1 About the Security Framework

Oracle Fusion Middleware and Oracle Real-Time Decisions share a common security framework. Using a common security framework enables Oracle Real-Time Decisions to interoperate securely within your Oracle Fusion Middleware deployment. The security framework is built upon the Java security model, which is a role-based, declarative model employing container-managed security where resources are protected by roles that are assigned to users.

For a more thorough discussion of the concepts discussed in this topic, see *Oracle Fusion Middleware Application Security Guide*.

Oracle Platform Security Services

Oracle Platform Security Services (OPSS) is the underlying platform on which the security framework is built. OPSS is standards-based and complies with role-based-access-control (RBAC), Java Enterprise Edition (JavaEE), and Java Authorization and Authentication Servers (JAAS).

Oracle WebLogic Server

Oracle Real-Time Decisions authentication is handled by the Oracle WebLogic Server authenticator providers, in compliance with the OPSS model. An authentication provider performs the following functions:

- Establishes the identity of users and system processes
- Transmits identity information
- Serves as a repository for identity information from where components can retrieve it

The default authentication provider is the directory server embedded in Oracle WebLogic Server. Alternate authentication providers can be used if desired and managed in the Oracle WebLogic Administration Console.

For more information on Oracle WebLogic Server authentication providers, see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

Oracle WebLogic Server Security Realms

An Oracle WebLogic Server security realm is specific to a domain, and contains the authentication providers, users, groups, security roles, and security policies configured together. Whereas multiple security realms can be defined for a domain, only one can be active, that is, designated as the default realm, at a given time.

Security Administration Tools

The administrative tasks required to secure and protect application objects are performed through Oracle Fusion Middleware and Oracle WebLogic Server consoles, and the command-line Oracle WebLogic Scripting Tool (WLST). For details, see Section 4.4, "Administration Tools Used for Common Security-Related Tasks."

4.2 Getting Started with Security for Oracle RTD

The security platform depends on certain key elements and processes to provide uniform security and identity management for all Oracle Fusion Middleware products. The default elements created during a simple install of Oracle RTD are used to illustrate this overview of security as it affects Oracle RTD users.

For more information about these elements, processes, and the security platform, see *Oracle Fusion Middleware Application Security Guide*.

4.2.1 The Security Controls for Oracle RTD

This topic introduces the security controls that relate to Oracle RTD, and the security configuration that is created during a default installation.

The key protections required for applications, and the basic questions they address, are:

- **Authentication**

  Who are the users allowed to access the application?

  Users and groups are stored in an identity store.

- **Authorization**

  What are the authenticated users allowed to do in and with the application?

  The roles and permissions allocated to authenticated users and groups of users are stored in a policy store.

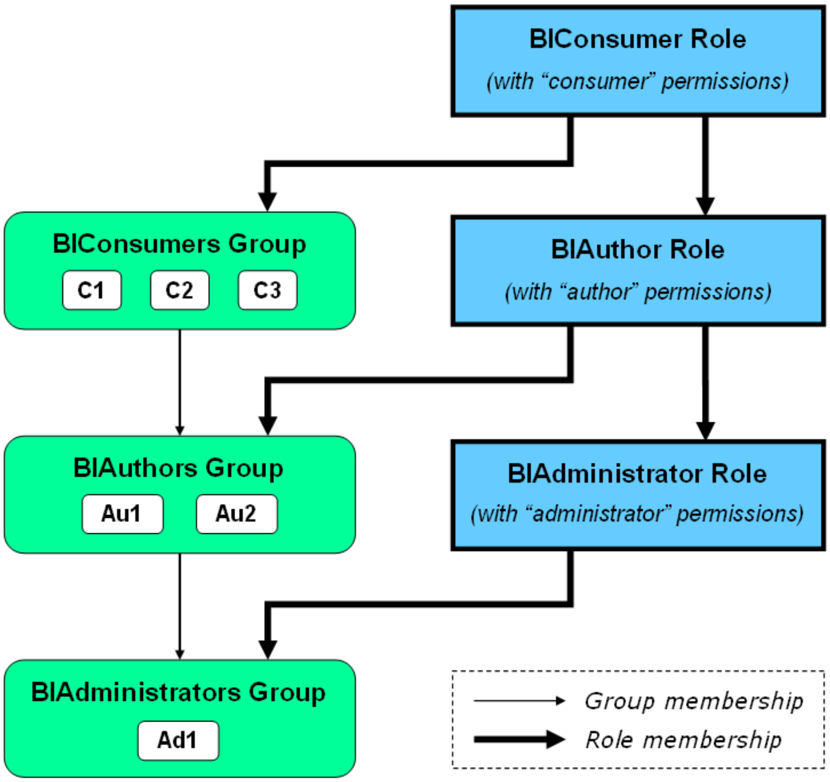
Table 4–1 summarizes the standard security controls for Oracle RTD.

Table 4–1 Standard Security Controls for Oracle RTD

Security Control	Main Purpose	Description
Identity store	Authentication	Trusted store to hold user and group identities.
Policy Store	Authorization	Trusted store used to hold the application roles and application grants that enable access to application objects.

To illustrate the security concepts, Figure 4–1 shows an example of the relationships between users, groups, application roles, and permissions, as defined and used in Oracle Fusion Middleware applications. This example is used as a reference point in subsequent descriptions of the individual security elements.

Figure 4–1 Example of Oracle Fusion Middleware Security Elements



The groups **BIConsumers**, **BIAuthors**, and **BIAdministrators**, and the application roles **BIConsumer**, **BIAuthor**, and **BIAdministrator**, are set up during installations that configure Oracle Real-Time Decisions or other Oracle Business Intelligence components. C1, C2, C3, Au1, Au2, Ad1 are examples of users who would be defined as members of their groups after installation.

By their membership in groups that are assigned to roles, users can inherit permissions from higher levels of group and role hierarchies.

For example, the authors Au1 and Au2 have two sets of permissions:

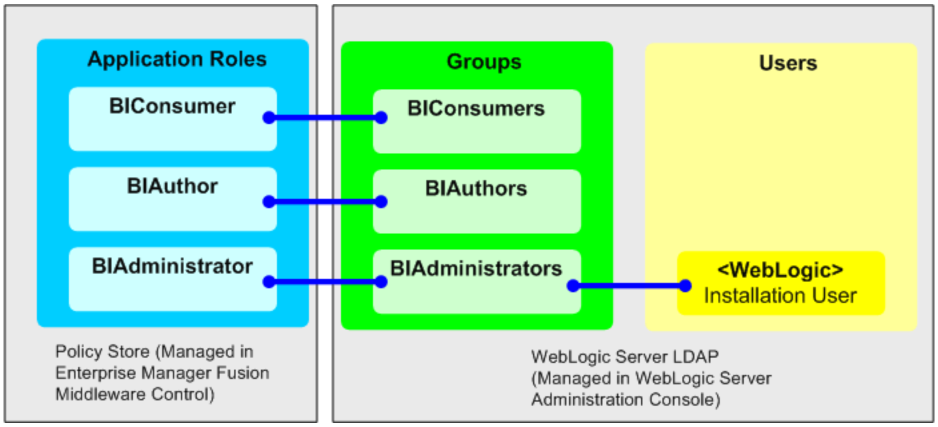
- Explicit permissions from the BIAuthor role, as the BIAuthors group is a member of the BIAuthor role

- Implicit permissions from the BIConsumer role, inherited through both the BIAuthor role and also through the BIConsumers group

The installed application roles are preconfigured with appropriate permissions and privileges to enable them to work with the installed Oracle BI and Oracle RTD software components. For example, the application role named BIAuthor is preconfigured with permissions and privileges to deploy Inline Services, and the application role BIConsumer allows access to Decision Center reports. For details of the default Oracle RTD privileges, see Section 4.3.1, "Default Oracle Real-Time Decisions Application Grants."

Figure 4–2 shows application roles, groups and users that are preconfigured during installation.

Figure 4–2 Preconfigured Application Roles, Groups, and Users


Application Roles
BIConsumer
BIAuthor
BIAdministrator
Groups
BIConsumers
BIAuthors
BIAdministrators
Users
<WebLogic>
Installation User
Policy Store (Managed in Enterprise Manager Fusion Middleware Control)
WebLogic Server LDAP (Managed in WebLogic Server Administration Console)

The user that is specified at installation time (for example, WebLogic), is automatically assigned to the WebLogic Administrators group named BIAdministrators and to the associated application role named BIAdministrator. The user has permissions to create and administer other users.

The rest of this section describes how users acquire their privileges to access applications and to control what they can do in the applications.

4.2.2 Key Authentication Elements

This section describes the security elements used for authentication.

In general, users and groups are defined in an identity store. User and group identities are stored in a directory server. Authentication of users and groups is performed by the authentication provider specified as part of Oracle WebLogic Server security setup.

Identity Store

An **identity store** contains the definitions of users, groups, and group hierarchies. Oracle WebLogic Server's embedded LDAP server is the default identity store. By default, the authentication provider DefaultAuthenticator authenticates against the users and groups in this LDAP server.

Oracle RTD can be reconfigured to use alternative directory servers, such as other LDAP servers. For a complete list, see *System Requirements and Supported Platforms for Oracle Fusion Middleware 11gR1*.

Users and Groups

A **user** is an entity that can be authenticated. A user can be a person, such as an application end user, or a software entity, such as a client application. Every user is given a unique identifier within the WebLogic domain where Oracle Real-Time Decisions is deployed. Every user has a unique identifier in the identity store and is therefore recognized across Oracle Fusion Middleware, Oracle WebLogic Server, and Oracle Real-Time Decisions.

Groups are created by organizing collections of users, and possibly other groups, who have something in common. Users can be defined in more than one group. A group is static identifier that is assigned by a system administrator.

> **Note:** By themselves, groups and groups hierarchies do not enable any privilege to perform any action within an application. Those privileges are conveyed through application roles and permissions, as described in Section 4.2.3, "Key Authorization Elements."

Default Identity Store

The **default identity store** is the LDAP-based embedded directory server provided by Oracle WebLogic Server, and managed using Oracle WebLogic Server Administration Console. It contains the default users and groups created during installation.

The default authentication provider is DefaultAuthenticator.

Default Users

In addition to two system users required for internal Oracle Fusion Middleware process management, there is a user with administrative privileges, whose name is entered during the installation.

> **Note:** For convenience, the name entered during installation is referred to as *<orig_admin_user>* in this section.

In the default security configuration, *<orig_admin_user>* is a member of the BIAdministrators group.

The default administrator user name *<orig_admin_user>* can be changed to a different value, and additional user names can be added by an administrative user using Oracle WebLogic Server Administration Console.

Default Groups

Table 4–2 lists the **default groups** and group members in the default identity store. These defaults can be changed to different values and additional group names can be added by an administrative user using Oracle WebLogic Server Administration Console.

Table 4–2 Default Groups and Members

Group Name	Group Members
BIAdministrators	any *administrative_user*
BIAuthors	BIAdministrators group
BIConsumers	BIAuthors group Oracle WebLogic Server LDAP server users group

These default group names serve as a starting point, by defining three broad categories of functional usage - administrator, author, and consumer - that correspond to the typical software user categories of administrator, application developer, and end-user. As indicated by Table 4–2 and the group hierarchy in Figure 4–1, an author is also considered to be a consumer, and an administrator is considered to be an author.

4.2.3 Key Authorization Elements

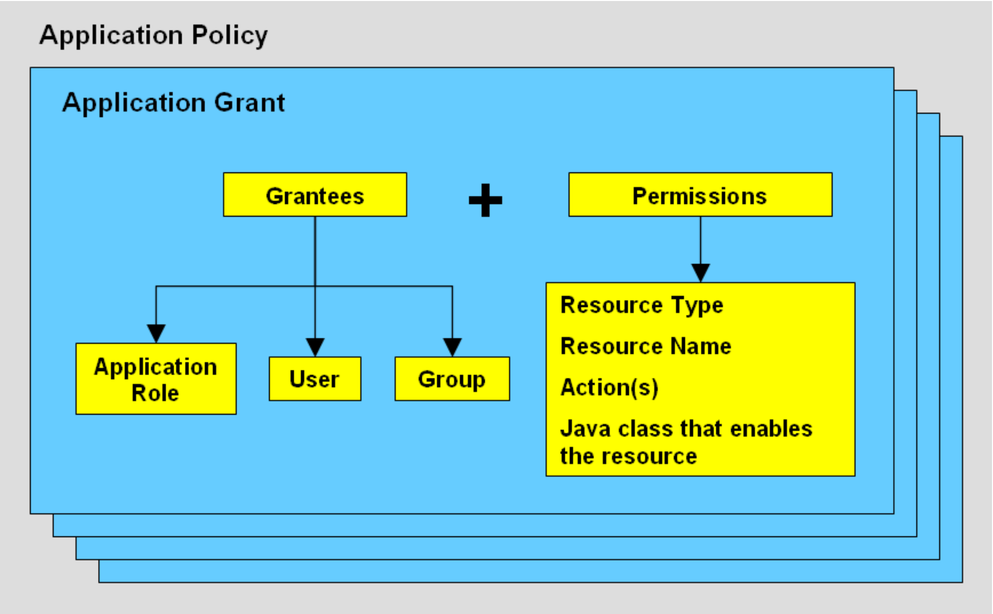
This section describes the security elements used for authorization.

Application Policy

An **application policy** is a collection of Java 2 and JAAS policies that are applicable to a specific application. The application policy defines who can do what on which application resources, and consists of one or more application grants.

Figure 4–3 shows a conceptual overview of the elements of an application policy. Descriptions of the individual components follow later in this section.

Figure 4–3 Application Policy Schematic Overview



Note: An **application stripe** defines a subset of policies in the policy store. The general application stripe used by all Oracle Business Intelligence components, including Oracle Real-Time Decisions, is named **obi**.

Application Role

An **application role** is a grouping construct in a policy store, that defines a collection of users and groups that need to perform a common set of application functions or processes. In general, an application role consists of users, groups, and other application roles.

Application roles provide the main way that permissions are given to application users. By themselves, application roles do not enable access to application objects - that is provided by mapping application roles to permissions in application grants in application policies.

Application Grant

An **application grant** is a combination of one or more grantees - each of which can be an application role, a group, or a user - and one or more permissions. For more information about users and groups, see Section 4.2.2, "Key Authentication Elements."

Permission

A **permission** is an extension of the Java permission concept, and consists of a Java class, a resource, and one or more actions allowed by the type of the resource.

For details and examples of the resources and actions available for Oracle RTD, see Section 4.3, "Resource Types and Actions for Oracle RTD."

Application Role Mapping

Any user or group assigned to an application role is granted the permissions associated with that role. More than one user or group can be assigned to the same application role.

Application role mapping is the process by which users, groups, and other application roles are dynamically mapped to application roles at runtime. Permissions are granted to users and groups according to which application roles they are members of, that is, have been mapped to.

Group and role hierarchies also illustrate the principle of inheritance: roles inherit other roles through the role hierarchy, and permissions are inherited through the group and role hierarchy. See Figure 4–1 for an example of the relationships between users, groups, application roles, and permissions.

Following the Figure 4–1 example, user **Au1** has all the permissions of the roles **BIAuthor** and **BIConsumer**, and user **Ad1** has all the permissions of the roles **BIAdministrator**, **BIAuthor**, and **BIConsumer**.

Policy Store

The **policy store** is the repository of system and application-specific policies. A policy store can be file-based or LDAP-based.

The default policy store is the `system.jazn-data.xml` file.

Default Policy Store

The **default policy store**, `system.jazn-data.xml`, contains the Oracle RTD policies, application roles, application grants, and default membership definitions as configured during installation.

Default Application Roles

Table 4–3 lists the **default application roles** and role members after installation. These defaults can be changed to different values and additional role names can be added by an administrative user using Oracle Fusion Middleware Control.

A graphical interpretation of these default application roles and the default group and role hierarchies appears in Figure 4–1, "Example of Oracle Fusion Middleware Security Elements".

Table 4–3 Default Application Roles and Role Members

Application Role Name	Role Members
BIAdministrator	BIAdministrators group

Table 4–3 (Cont.) Default Application Roles and Role Members

Application Role Name	Role Members
BIAuthor	BIAuthors group BIAdministrator application role
BIConsumer	BIConsumers group BIAuthor application role

The **BIAdministrator** role is intended for administrative permissions necessary to configure and manage the Oracle RTD installation. Any member of the BIAdministrators group is explicitly granted this role and implicitly granted the BIAuthor and BIConsumer roles. See Table 4–5 for a list of the default Oracle RTD application grants for this role.

The **BIAuthor** role is intended for permissions necessary to create and edit content for others to consume. Any member of the BIAuthors group is explicitly granted this role and implicitly granted the BIConsumer role. See Table 4–5 for a list of the default Oracle RTD application grants for this role.

The **BIConsumer** role is intended for permissions necessary to consume content created by others. See Table 4–5 for a list of the default Oracle RTD application grants for this role.

> **Note:** The specialized role **authenticated_role** is granted by default to any authenticated user. It is a member of the **BIConsumer** role by default. Removal of authenticated_role would result in the inability to log into the system. For more information, see "The Authenticated Role" in *Oracle Fusion Middleware Application Security Guide*.

Default Application Grants

The default application grants for Oracle RTD users after installation are described in Section 4.3.1, "Default Oracle Real-Time Decisions Application Grants."

4.3 Resource Types and Actions for Oracle RTD

OPSS includes the Java class `oracle.security.jps.ResourcePermission` that can be used as the permission class within any grant to protect application or system resources. Oracle RTD uses this class to control access to three types of resource:

- Inline Service
- Decision Center Perspective
- Batch Job

Table 4–4 shows the resource types supported by Oracle RTD and their associated actions.

Table 4–4 Oracle RTD Resource Types and Actions

Type of Resource	Resource Type Name Stored in Application Grants	Action[:Qualifier]	Comments
Inline Service	rtd_ils	choice_editor	May execute any methods of the **ExternalChoice** web service for the named Inline Service.
		decision_service:normal	May execute any integration points (advisors and informants) for the named Inline Service. Action qualifier **normal** allows integration point requests to be executed in the server.
		decision_service:stress	May execute any integration points (Advisors and Informants) for the named Inline Service. Action qualifier **stress** allows LoadGen to issue integration point calls. To be accepted by the server, the user also needs the **normal** action.
		open_service:read	Authorizes the use of Decision Center to open the named Inline Service for viewing. Also authorizes the External Rule Editor to access the named Inline Service, since the External Rule Editor does not need to update the content of the Inline Service.
		open_service:write	Authorizes the use of Decision Center to open the named Inline Service for editing.
		deploy_service	Authorizes the deployment of the named Inline Service from Decision Studio.
		download_service	Authorizes the use of Decision Studio to download the named Inline Service from a server.
		clear_choice_history	Authorizes the clearing of the choice history for the named Inline Service through the Administration web service.
		clear_study	Authorizes the clearing of the study for the named Inline Service through the Administration web service. *A study is not shared by multiple Inline Services, so naming the owning Inline Service is equivalent to naming the study.*
		clear_statistics	Authorizes the clearing of the statistics for the Inline Service through the Administration web service.
		clear_model	Authorizes the clearing of the model for the named Inline Service through the Administration web service.
		clear_all_operational_data	Authorizes the clearing of the operational data for the named Inline Service through the Administration web service.
		delete_service	Authorizes the deletion of the named Inline Service through the Administration web service.
		unlock_service	Authorizes the unlocking of the named Inline Service through the Administration web service.
Decision Center Perspective	rtd_dc_persp	dc_perspective	Open the named Decision Center Perspective, to have Decision Center render its specialized set of UI elements or capabilities.
Registered Batch Job Type	rtd_batch	batch_admin	May execute any methods of the BatchManager web service to start, stop, or query the status of the registered batch job type name.

4.3.1 Default Oracle Real-Time Decisions Application Grants

The default file-based policy store includes pre-configured application grants. Oracle RTD uses the permission class, `oracle.security.jps.ResourcePermission`, which references attributes of the resource types listed in Table 4–4.

Table 4–5 lists the default application roles, Oracle RTD resource types, resource names, and actions in the default application grants after installation.

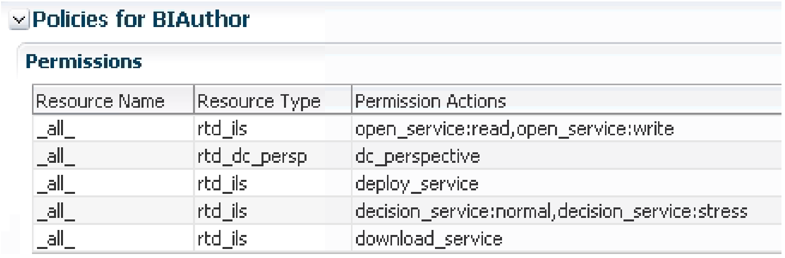
Note: The resource name **_all** _ is a special name that matches any Oracle RTD resource name of the associated resource type.

Table 4–5 Default Application Grants for Oracle RTD Users

Application Role	Resource Type	Resource Name	Action[:Qualifier]
BIAdministrator	rtd_ils	_all_	open_service:read
		all	open_service:write
		all	deploy_service
		all	download_service
		all	choice_editor
		all	decision_service:normal
		all	decision_service:stress
		all	clear_choice_history
		all	clear_study
		all	clear_statistics
		all	clear_model
		all	clear_all_operational_data
		all	delete_service
		all	unlock_service
	rtd_dc_persp	_all_	dc_perspective
	rtd_batch	_all_	batch_admin
BI Author	rtd_ils	_all_	open_service:read
		all	open_service:write
		all	deploy_service
		all	download_service
		all	decision_service:normal
		all	decision_service:stress
	rtd_dc_persp	_all_	dc_perspective
BI Consumer	rtd_ils	_all_	open_service:read
		all	choice_editor
		all	decision_service:normal
	rtd_dc_persp	Explore	dc_perspective
		At a Glance	dc_perspective
	rtd_batch	_all_	batch_admin

In the Fusion Middleware Control Application Policies screen, the default permissions related to Oracle RTD for the application role BIAuthor appear as shown in Figure 4–4.

Figure 4–4 Example of Permissions in Fusion Middleware Control


Policies for BIAuthor

Permissions

Resource Name	Resource Type	Permission Actions
all	rtd_ils	open_service:read,open_service:write
all	rtd_dc_persp	dc_perspective
all	rtd_ils	deploy_service
all	rtd_ils	decision_service:normal,decision_service:stress
all	rtd_ils	download_service

For details of how to create and edit application roles and application policies, see Section 4.7.4, "Managing the Policy Store Using Fusion Middleware Control."

4.4 Administration Tools Used for Common Security-Related Tasks

Oracle Real-Time Decisions shares a common security framework with the Oracle Fusion Middleware platform. This common security configuration utilizes Oracle WebLogic Server as the defacto administration server. The implementation details are largely hidden while performing daily administrative tasks and are exposed only by the tools used to manage your Oracle Real-Time Decisions security configuration. The two main administration tools are:

- **Oracle WebLogic Server Administration Console** is used to manage users and groups for the embedded LDAP server that serves as the default identity store after a default Simple install
- **Oracle Enterprise Manager Fusion Middleware Control** is used to define application roles and application policies that grant permissions to users, groups, or other application roles

In addition, the Oracle WebLogic Scripting Tool (WLST) is a command-line scripting tool that you can use to create, manage, and monitor Oracle WebLogic Server domains, and administer Oracle Fusion Middleware security features. For more information about using WLST, see *Oracle Fusion Middleware Oracle WebLogic Scripting Tool* and *Oracle Fusion Middleware WebLogic Scripting Tool Command Reference*.

Table 4–6 lists common security-related tasks performed and the administration tool used.

Table 4–6 Common Tasks and Administration Tool Used

Task	Tool to Use
Manage Users and Groups for Authentication	Oracle WebLogic Server Administration Console
Manage Application Roles and Application Policies	Fusion Middleware Control

4.5 Typical System Administration Tasks for Securing Oracle RTD

Table 4–7 shows the typical system administration tasks that you perform to secure Oracle RTD and where to find related information.

Table 4–7 Typical System Administration Tasks Performed to Secure Oracle RTD

Task	For More Information
Managing users and groups for authentication	Section 4.6, "Managing Authentication for Oracle RTD"
Granting privileges to access Oracle RTD resources	Section 4.7, "Managing Authorization and Privileges for Oracle RTD"
Decide if using SSL in your deployment	Section 4.8, "Using SSL with Oracle RTD"
Enabling SSO authentication	Section 4.9, "Enabling SSO Authentication"

4.6 Managing Authentication for Oracle RTD

This section contains the following topics:

- Section 4.6.1, "Task Map: Configuring Authentication for Oracle RTD"
- Section 4.6.2, "Understanding Oracle Real-Time Decisions Authentication"
- Section 4.6.3, "Managing the Default Authentication Provider"
- Section 4.6.4, "Configuring a New Authentication Provider"

> **Note:** For configuring authentication using a Single Sign-On solution, see "Introduction to Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*.

4.6.1 Task Map: Configuring Authentication for Oracle RTD

The following task map contains common authentication configuration tasks and provides links for obtaining more information.

Task	Description	For Information
Decide on authentication method	Decide whether to use the default embedded directory server (LDAP-based) or a different external authentication method	Section 4.6.2, "Understanding Oracle Real-Time Decisions Authentication"
Configure the default authentication provider	Configure the default authentication provider for the default security realm.	Section 4.6.3, "Managing the Default Authentication Provider"
Add users and groups	Add users and groups to the identity store	Section 4.6.3.1, "Managing Users and Groups"
Configure an alternate authentication provider to authenticate users	Configure an alternate authentication provider.	Section 4.6.4, "Configuring a New Authentication Provider"

4.6.2 Understanding Oracle Real-Time Decisions Authentication

During installation an Oracle WebLogic Server domain is created and Oracle Real-Time Decisions is installed into that domain. Security for an Oracle WebLogic Server domain is managed in context of the domain's security realm. A **security realm** acts as a scoping mechanism. Each security realm consists of a set of configured security providers, users, groups, security roles, and security policies. Only one security realm can be active for the domain.

Oracle Real-Time Decisions authentication is performed by the authentication provider configured for the default security realm for the WebLogic Server domain in which Oracle Real-Time Decisions is installed. Oracle WebLogic Server Administration Console is the administration tool for managing an Oracle WebLogic Server domain.

The following sections include a brief introduction to key Oracle WebLogic Server security concepts. For more information about Oracle WebLogic Server security and how it is managed, see *Oracle Fusion Middleware Understanding Security for Oracle WebLogic Server* and *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

4.6.2.1 Identity Stores and Authentication Providers

An **identity store** contains user name, password, and group membership information. It serves as the data store for user credentials. An **authentication provider** accesses the stored user information and is responsible authenticating a user. For example, when a user name and password combination is entered at log in, the authentication provider searches the identity store to verify the credentials provided. If SSO authentication is configured for Oracle RTD, the SSO provider also use the data contained in this identity store.

If using an identity store other than the embedded directory server included with Oracle WebLogic Server, the default users and groups shown in Section 4.2.2, "Key Authentication Elements" will not be automatically present. You can create users and groups with names of your own choosing or re-create the default user and group names if the authentication provider supports this. After this work is completed, you must map the default Oracle RTD application roles the equivalent groups. For example, if your corporate LDAP server is being used as the identity store and you are unable to re-create the Oracle RTD default users and groups in it, you will need to map the default application roles to different groups specific to the corporate LDAP server. For more information about the default application roles and group mappings, see Section 4.2.2, "Key Authentication Elements" and Section 4.2.3, "Key Authorization Elements."

4.6.3 Managing the Default Authentication Provider

After installation, Oracle Real-Time Decisions is configured to use the Oracle WebLogic Server default authentication provider (DefaultAuthenticator). DefaultAuthenticator supports user name and password authentication. The Oracle WebLogic Server embedded directory server is configured as the default user data source (identity store). While validating authentication requests, the authentication provider connects to the identity data store to verify credentials. The authentication provider uses the user data store configured in Oracle WebLogic Server Administration Console.

The active security realm can have multiple authentication providers configured but only one provider can be active at a time. The order of providers in the list determines priority. The effect of having multiple authentication providers defined in a security realm is not cumulative; rather, the first provider in list is the source for all user and password data needed during authentication. Having the ability to define more than one authentication provider enables you to switch between authentication providers by rearranging order in the list. For example, if you have separate directory servers for your development and production environments, you can change which server is to be used during authentication by re-ordering them in the list.

Detailed information about managing and configuring an authentication provider in Oracle WebLogic Server is available in its online help. For more information, log into Oracle WebLogic Server Administration Console and launch *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

4.6.3.1 Managing Users and Groups

Groups are logically ordered sets of users. Managing a group is more efficient than managing a large number of users individually. The best practice is to first organize all Oracle RTD users into groups that have similar system access requirements. Application roles that provide the correct level of access then can be mapped to these groups. If system access requirements change then you need only modify the permissions granted by the application roles, or create a new application roles with appropriate permissions. After your groups are established, continue to add or remove users directly in the user data source (identity store) using its administration interface as you normally would.

The default identity store is Oracle WebLogic Server embedded directory server. But there are many other supported directory servers that can be used, as well as alternative sources such as a database or a table. For information about adding users or groups to a non-default directory server, consult that product's documentation. For a current list of supported authentication providers and directory servers to use with Oracle RTD, see the system requirements and certification documentation. For more information, see Section 1.4, "System Requirements and Certification."

For more information about managing users and groups in the default directory server, see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

To create a user in the default directory server:

1. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.
2. Select **Users and Groups** tab, then **Users**. Click **New**.
3. In the **Create a New User** page provide the following information:
   - **Name**: Enter the name of the user. See online help for a list of invalid characters.
   - (Optional) **Description**: Enter a description.
   - **Provider**: Select the authentication provider from the list that corresponds to the data store where the user information is contained. DefaultAuthenticator is the name for the default authentication provider.
   - **Password**: Enter a password for the user that is at least 8 characters long.
   - Confirm Password: Re-enter the user password.
4. Click **OK**.

   The user name is added to the User table.

To create a group in the default directory server:

1. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.
2. Select **Users and Groups** tab, then **Groups**. Click **New**.

3. In the **Create a New Group** page provide the following information:
   - **Name**: Enter the name of the Group. Group names are case insensitive but must be unique. See online help for a list of invalid characters.
   - (Optional) **Description**: Enter a description.
   - **Provider**: Select the authentication provider from the list that corresponds to the data store where the group information is contained. DefaultAuthenticator is the name for the default authentication provider.
4. Click **OK.**

   The group name is added to the Group table.

To add a user to a group in the default directory server:

1. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.
2. Select **Users and Groups** tab, then **Users**.
3. In the Users table select the user you want to add to a group.
4. Select the **Groups** tab.
5. Select a group or groups from the **Available** list box.
6. Click **Save**.

To change a user password in the default directory server:

1. In Oracle WebLogic Server Administration Console, select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.
2. Select **Users and Groups** tab, then **Users**
3. In the Users table select the user you want to change the password for.
4. Select the **Passwords** tab and enter the password in the **New Password** and **Confirm Password** fields.
5. Click **Save**.

4.6.4 Configuring a New Authentication Provider

You have the option to use several different types of authentication providers in any environment, such as an LDAP server or a database. Configuring Oracle RTD to use an alternative external identity store is performed using the Oracle WebLogic Server Administration Console.

For a current list of supported authentication providers and directory servers to use with Oracle RTD, see the system requirements and certification documentation. For more information, see Section 1.4, "System Requirements and Certification".

Any identity store provider supported by Oracle WebLogic Server can be configured to be used with Oracle RTD. Oracle RTD delegates authentication and user population management to the authenticator and identity store configured for the domain in which it is deployed. For example, if configured to use Oracle WebLogic Server's default authenticator, then management is performed in the Oracle WebLogic Server Administration Console. If configured to use Oracle Internet Directory (OID), then the OID management user interface is used, and so on.

Note: If a directory server other than the default is being used as the user data source for the new authentication provider, you will still be able to view the users and groups from that directory server in Oracle WebLogic Server Administration Console. However, you will continue to manage the users and groups in the interface for the directory server.

Oracle RTD uses the first authentication provider configured for the active security realm in the WebLogic Server domain. The active security realm for the domain can have multiple authentication providers configured. Their order in the list determines their priority. The effect of multiple authentication providers is not cumulative; rather, the provider in the first position is the source for all user and password definitions required for authentication. This allows you to switch between authentication providers as needed. For example, if you have separate LDAP servers for your development and production environments, you can change which server is used for authentication by re-ordering them.

If using an identity store provider other than the one installed as part of the default security configuration, the default users and groups discussed in Section 4.2.2, "Key Authentication Elements" will not be automatically present. You can create users and groups with names of your own choosing or re-create the default user and group names if the authentication provider supports this. After this work is completed, you must map the default Oracle RTD application roles to different groups again. For example, if your corporate LDAP server is being used as the identity store and you are unable to re-create the Oracle RTD default users and groups in it, you will need to map the default application roles to different groups specific to the corporate LDAP server.

Note: If the security realm is configured to use an authentication provider other than the default embedded LDAP server, the application roles must be mapped again to the correct groups (enterprise roles) in the alternative identity store.

To configure the authentication security provider, log into Oracle WebLogic Server Administration Console and see the detailed steps in *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*. For information about configuring Oracle Internet Directory as an authentication provider, see Section 4.6.4.1, "Configuring Oracle Internet Directory as an Authentication Provider."

4.6.4.1 Configuring Oracle Internet Directory as an Authentication Provider

Oracle Internet Directory is used in the following procedures to explain the process of configuring a different authentication provider and identity store combination. Using the same directory server for both is convenient; however, you can use any combination of directory servers as long as they are both supported by Oracle RTD.

Configuring Oracle Internet Directory to be both the authentication provider and identity store demonstrates the process but differences will exist with another directory server is used. For additional information about configuring an authentication provider for an Oracle WebLogic Server domain, see *Oracle Fusion Middleware Oracle WebLogic Server Administration Console Online Help*.

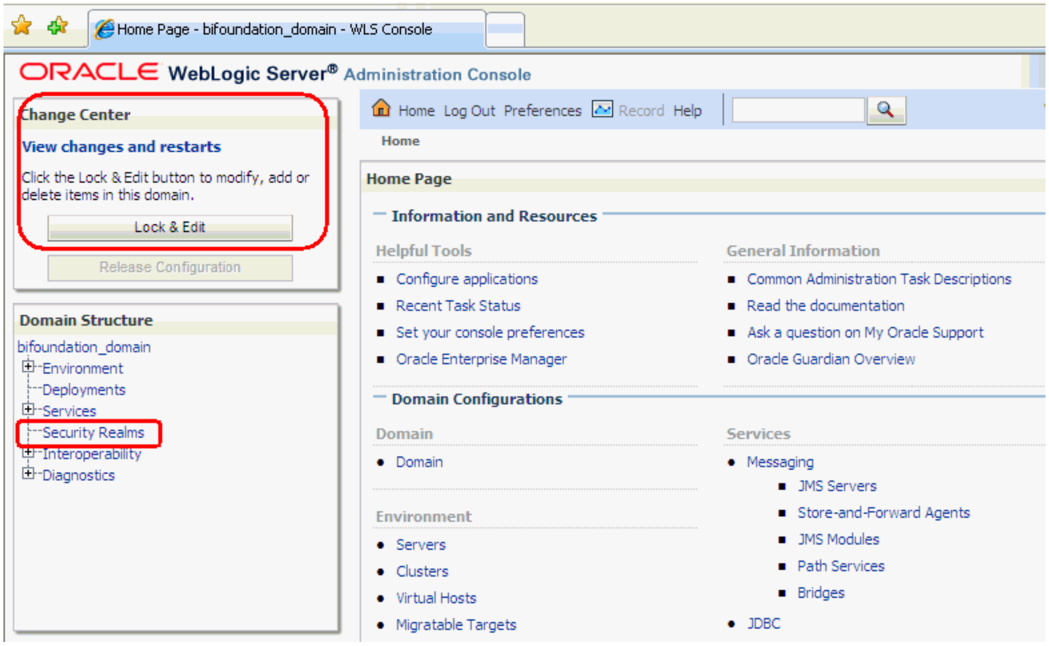
The Oracle Internet Directory authentication provider is configured in the Administration Console when Oracle Internet Directory provides the user data (identity store).

The rest of this section describes how to configure the Oracle Internet Directory authentication provider, and how to reorder the authentication provider list.

To configure the Oracle Internet Directory authentication provider:

In the following description, MyOIDDirectory is used to represent the Oracle Internet Directory.

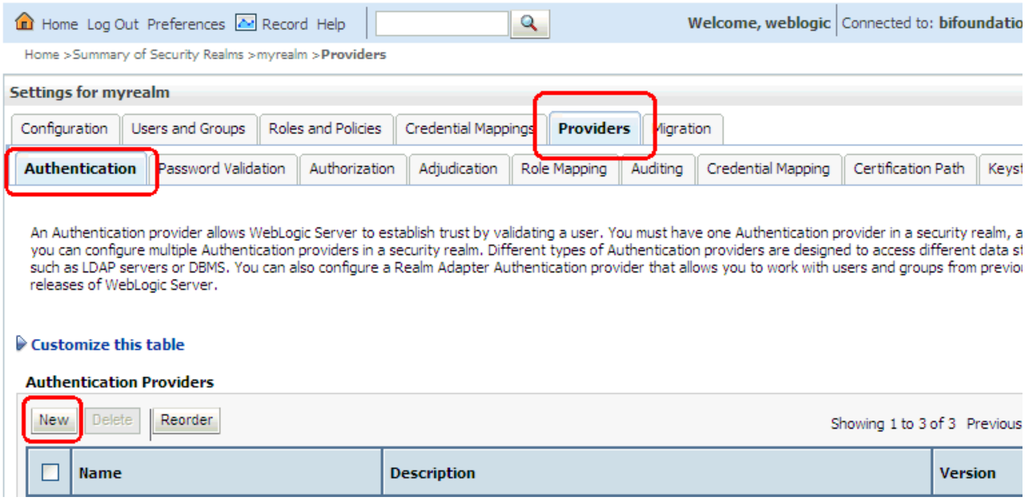
1. Click **Lock & Edit** in the Change Center of the Oracle WebLogic Server Administration Console.



Click the Lock & Edit button to modify, add or delete items in this domain.

2. Select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**.

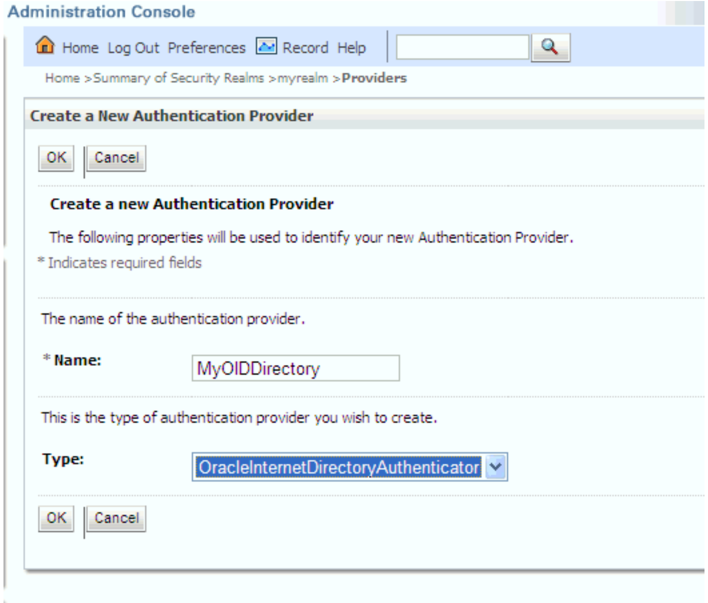
3. Select **Providers**, then **Authentication**. Click **New** to launch the **Create a New Authentication Provider** page.



An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security realm, a you can configure multiple Authentication providers in a security realm. Different types of Authentication providers are designed to access different data st such as LDAP servers or DBMS. You can also configure a Realm Adapter Authentication provider that allows you to work with users and groups from previo releases of WebLogic Server.

4. Enter values in the **Create a New Authentication Provider** page as follows:

   - **Name**: Enter a name for the authentication provider. For example, MyOIDDirectory.
   - **Type**: Select OracleInternetDirectoryAuthenticator from the list.
   - Click **OK**.



Home >Summary of Security Realms >myrealm >**Providers**

Create a New Authentication Provider

Create a new Authentication Provider

The following properties will be used to identify your new Authentication Provider.

* Indicates required fields

The name of the authentication provider.

* **Name:** MyOIDDirectory

This is the type of authentication provider you wish to create.

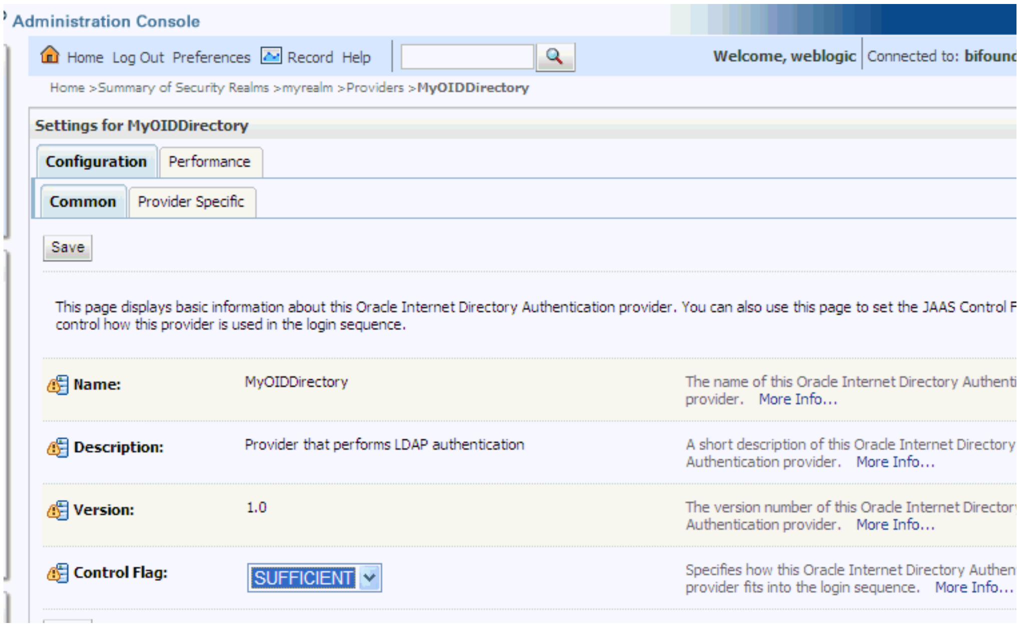
Type: OracleInternetDirectoryAuthenticator

5. Select **Providers**, then **Authentication**. Click the name of the authentication provider to complete its configuration. For example, MyOIDDirectory.

   The **Configuration** page for the Oracle Internet Directory authentication provider is displayed and has multiple tabs.For more information about completing fields in the **Configuration** page, click the **More Info...** link located in each field.

   You next set the Control Flag for the Oracle Internet Directory authentication provider. When configuring multiple authenticator providers, the Control Flag controls how the authentication providers are used in the login sequence.

6. On the Common tab, set the **Control Flag** to SUFFICIENT by selecting it from the list. Click **More Info...** for more information about the Control Flag settings.



Home >Summary of Security Realms >myrealm >Providers >MyOIDDirectory

Settings for MyOIDDirectory

This page displays basic information about this Oracle Internet Directory Authentication provider. You can also use this page to set the JAAS Control F control how this provider is used in the login sequence.

Name:	MyOIDDirectory	The name of this Oracle Internet Directory Authenti provider. More Info...
Description:	Provider that performs LDAP authentication	A short description of this Oracle Internet Directory Authentication provider. More Info...
Version:	1.0	The version number of this Oracle Internet Director Authentication provider. More Info...
Control Flag:	SUFFICIENT	Specifies how this Oracle Internet Directory Authen provider fits into the login sequence. More Info...

7. Select the **Provider Specific** tab and complete these fields as follows. Click **More Info...** for information about completing the additional fields in each section.

Section Name	Field Name	Description
Connection	Host	The host name of the Oracle Internet Directory server.
	Port	The port number on which the Oracle Internet Directory server is listening.
	Principal	The distinguished name (DN) of the Oracle Internet Directory user to be used to connect to the Oracle Internet Directory server. For example: cn=OIDUser,cn=users,dc=us,dc=mycompany,dc=com
	Credential	Password for the Oracle Internet Directory user entered as the Principal.
Users	User Base DN	The base distinguished name (DN) of the Oracle Internet Directory server tree that contains users.
Groups	Group Base DN	The base distinguished name (DN) of the Oracle Internet Directory server tree that contains groups.

8. Click **Save**.

9. Click **Activate Changes** in the Change Center.

   The Administration and Managed Servers must be restarted.

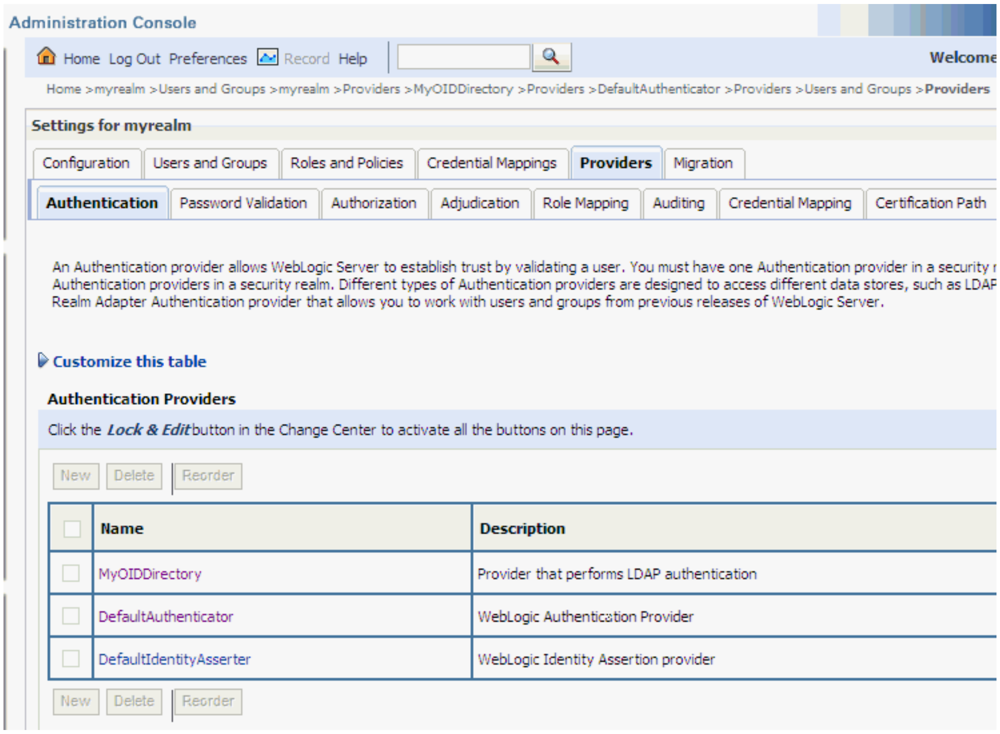
10. Restart Oracle WebLogic Server.

To reorder authentication providers:

The Authentication Providers page in Oracle WebLogic Server Administration Console lists all authentication providers configured for the default security realm. Oracle RTD uses only the authentication provider that is in the first position. If multiple authentication providers are configured, you must move to the first position the authentication provider that Oracle RTD is to use.

1. Click **Lock & Edit** in the Change Center of the Oracle WebLogic Server Administration Console.

2. Select **Security Realms** from the left pane and click the realm you are configuring. For example, **myrealm**. Select **Security Realms** from Domain Structure in the left pane.
3. Select the **Providers** tab, then **Authentication**.
4. Click **Reorder**.
5. Select the name of the Oracle Internet Directory authentication provider and use the arrow buttons to move it into the first position. Your results should resemble the following figure where MyOIDDirectory represents the Oracle Internet Directory.



An Authentication provider allows WebLogic Server to establish trust by validating a user. You must have one Authentication provider in a security r
Authentication providers in a security realm. Different types of Authentication providers are designed to access different data stores, such as LDAP
Realm Adapter Authentication provider that allows you to work with users and groups from previous releases of WebLogic Server.

Authentication Providers

Click the *Lock & Edit* button in the Change Center to activate all the buttons on this page.

□	Name	Description
□	MyOIDDirectory	Provider that performs LDAP authentication
□	DefaultAuthenticator	WebLogic Authentication Provider
□	DefaultIdentityAsserter	WebLogic Identity Assertion provider

4.7 Managing Authorization and Privileges for Oracle RTD

This section contains the following topics:

- Section 4.7.1, "Task Map: Configuring Authorization for Oracle RTD"
- Section 4.7.2, "Understanding the Authorization Process"
- Section 4.7.3, "Configuring the Policy Store"
- Section 4.7.4, "Managing the Policy Store Using Fusion Middleware Control"

4.7.1 Task Map: Configuring Authorization for Oracle RTD

This task map contains common authorization configuration tasks and provides links for more information.

Task	Description	Information
Decide on authorization method	Decide if the policy store will be the default file or LDAP-based	Section 4.7.2, "Understanding the Authorization Process"

Task	Description	Information
Configure a policy store	Configure and reassociate a policy store	Section 4.7.3, "Configuring the Policy Store"
Create, edit, and delete application roles and application policies	Create, edit, and delete application roles and application policies using Fusion Middleware Control	Section 4.7.4, "Managing the Policy Store Using Fusion Middleware Control"

4.7.2 Understanding the Authorization Process

After a user is authenticated, further access to Oracle RTD is controlled through the application grants in application policies in the policy store, which is managed by Fusion Middleware Control.

4.7.2.1 Policy Stores

The policy store contains the system and application-specific policies and roles used by an application. A domain policy store can be file-based or LDAP-based. The default policy store is installed as an XML file (`system-jazn-data.xml`). This XML file holds the mapping definitions between the default Oracle RTD application roles, permissions, users and groups all configured as part of installation.

Oracle RTD permissions are granted by mapping users and groups from the identity store to application roles and application grants located in the policy store. These mapping definitions between users and groups (identity store) and the application roles (policy store) are also kept in the policy store.

Both type of policy store, file-based and LDAP-based, are managed using Fusion Middleware Control.

4.7.3 Configuring the Policy Store

The default `system-jazn-data.xml` file is pre-configured as the default policy store during installation. You can continue to use the default and modify it as needed for your environment or you can migrate its data to an LDAP-based provider. An LDAP-based provider is typically used and recommended in production environments.

Permissions must be defined in a manner that Oracle RTD understands. All valid Oracle RTD resources types and resource names are provided and are pre-mapped to the default application roles, as described in Section 4.3, "Resource Types and Actions for Oracle RTD." You cannot create new resources types for Oracle RTD, but you can select a specific name for a resource instead of the dummy name "_all_".

Using the appropriate administration interface you can tailor the application grants for the application policy and role definitions contained in the policy store.

4.7.3.1 Configuring an LDAP-Based Policy Store

The only LDAP server supported in this release is Oracle Internet Directory. For more information, see "Using an LDAP-Based OPSS Security Store" in *Oracle Fusion Middleware Application Security Guide*.

4.7.3.2 Reassociating the Policy Store

Migrating policies and credentials from one security store to another is called reassociation. Policy store data can be reassociated from a file-based store to an LDAP-based store, or from an LDAP-based store to another LDAP-based store.

Reassociation is commonly done when moving from one environment to another, such as from a test environment to a production environment.

For more information about reassociation and the steps required to migrate policy store data to Oracle Internet Directory, see "Reassociating with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4 Managing the Policy Store Using Fusion Middleware Control

A policy store is managed using Fusion Middleware Control. For information about using Fusion Middleware Control, see *Oracle Fusion Middleware Administrator's Guide*.

Caution: As part of your standard backup strategy, you should make a copy of the original `system-jazn-data.xml` policy file and place it in a safe location. Use the copy of the original file to restore the default policy store configuration, if needed. Changes to the default security configuration may lead to an unwanted state.

The two main objects that control authorization are application roles and application policies. For general information about these objects and their inter-relationship with users and groups, see Section 4.2.3, "Key Authorization Elements" and Section 4.2.1, "The Security Controls for Oracle RTD."

The application roles, application grants, and groups that make up the default security configuration are pre-mapped to each other as detailed in Section 4.3.1, "Default Oracle Real-Time Decisions Application Grants."

Application Roles and Application Policies

An application role consists of users, groups, and other application roles. Users and groups are created in the identity store associated with the authentication provider, and can be assigned to application roles in Fusion Middleware Control.

Application grants, that control who can perform which operations on which resources, are defined in application policies in Fusion Middleware Control. An application grant typically consists of a set of application-oriented permissions and one or more application roles that are granted those permissions.

In addition to using the default application roles and application policies created during installation, you can create your own application roles and application policies. A simplified overview of the creation process is as follows:

1. Create an application role, and add one or more users, groups, and existing application roles to your new role.
2. Create an application policy, specifying one or more application-oriented permissions, together with one or more grantees. Typically the grantees are application roles, but in general a grantee can be an application role, a user, or a group.

   Only after you have added a role to an application policy will the role become effective in authorizing the permissions in the application policy.

In general, in Fusion Middleware Control, there are two methods for creating a new application role or an application policy:

- Create the application role or the application policy by explicitly defining their constituent elements.

- Create the application role or the application policy based on an existing application role or application policy: you copy the components from the existing object, then add or modify them.

> **Note:** Before creating a new application role to incorporate into your security configuration, familiarize yourself with how permission and group inheritance works. It is important when constructing an application role hierarchy that circular dependencies are not introduced.

The rest of this section describes how to manage application roles and application policies in Fusion Middleware Control, and contains the following topics:

- Section 4.7.4.1, "Creating a New Application Role"
- Section 4.7.4.2, "Creating an Application Role Like Another Application Role"
- Section 4.7.4.3, "Editing an Application Role"
- Section 4.7.4.4, "Deleting an Application Role"
- Section 4.7.4.5, "Creating a New Application Policy"
- Section 4.7.4.6, "Creating an Application Policy Like Another Application Policy"
- Section 4.7.4.7, "Editing an Application Policy"
- Section 4.7.4.8, "Deleting an Application Policy"

4.7.4.1 Creating a New Application Role

The following is an overview of the process to create a new application role:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Roles.
3. Select and search for application roles in the Application Stripe **obi** (click the button beside the Role Name box).
4. Click **Create**...
5. In the Create Application Role page:
   - Enter **Role Name**
   - Optionally enter Display Name and Description
   - Add one or more Application Roles, Groups, Users

     In each case, you can search and select from the available application roles, groups, and users.

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4.2 Creating an Application Role Like Another Application Role

The following is an overview of the process to create an application role by copying from another application role:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Roles.
3. Select and search for application roles in the Application Stripe **obi** (click the button beside the Role Name box).
4. Select an Application Role in the search results.
5. Click **Create Like...**
6. In the Create Application Role Like... page:
   - Change the **Role Name**
   - Optionally edit Display Name and Description
   - Add or delete one or more Application Roles, Groups, Users

     For each addition, you can search and select from the available application roles, groups, and users.

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4.3 Editing an Application Role

The following is an overview of the process to edit an application role:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Roles.
3. Select and search for application roles in the Application Stripe **obi** (click the button beside the Role Name box).
4. Select an Application Role in the search results.
5. Click **Edit...**
6. In the Edit Application Role page:
   - Optionally edit Display Name and Description
   - Add or delete one or more Application Roles, Groups, Users

     For each addition, you can search and select from the available application roles, groups, and users.

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4.4 Deleting an Application Role

The following is an overview of the process to delete an application role:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."

2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Roles.
3. Select and search for application roles in the Application Stripe **obi** (click the button beside the Role Name box).
4. Select an Application Role in the search results.
5. Click **Delete...**
6. Confirm that you want to delete the application role.

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4.5 Creating a New Application Policy

The following is an overview of the process to create a new application policy:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Policies.
3. Select and search for security grants in the Application Stripe **obi** (click the right arrowhead button beside the Name box).
4. Select an Application Policy in the search results.
5. Click **Create...**
6. In the Create Application Grant page:
   - Add, edit or delete one or more Permissions
   - Add, edit, or delete one or more Grantees

   When creating an application grant, you must add at least one permission and one grantee.

   Add One or More Permissions

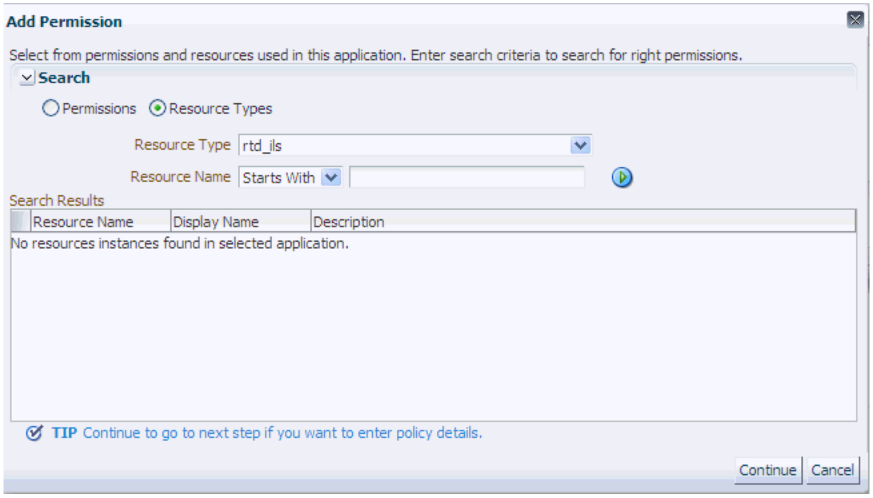
   The default permissions for Oracle RTD appear in Section 4.3, "Resource Types and Actions for Oracle RTD," and contain the dummy Resource Name "_all_" that matches any Oracle RTD resource name of the associated resource type.

   You can add permissions through either of the following processes:

   - Search for existing permissions in the permission class **oracle.security.jps.ResourcePermission** and optionally modify them
   - Specify permission actions for Oracle RTD resources, that is, resources that have one of the Resource Types specified in Section 4.3, "Resource Types and Actions for Oracle RTD"

   For example, to enable a single Inline Service Market_ILS to be opened for read and write and to be deployed, perform the following steps:

   - In the Add Permission window, select the radio button Resource Types, and for the Resource Type, select rtd_ils.



Add Permission

Select from permissions and resources used in this application. Enter search criteria to search for right permissions.

Search

○ Permissions ● Resource Types

Resource Type: rtd_ils

Resource Name: Starts With

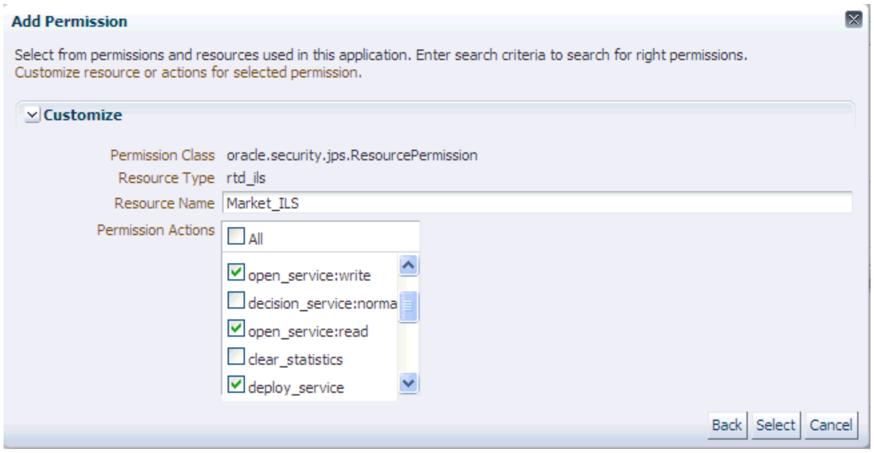
Search Results

Resource Name	Display Name	Description

No resources instances found in selected application.

TIP Continue to go to next step if you want to enter policy details.

- Click Continue.
- In the Customize area, enter Market_ILS for the Resource Name, and select the appropriate Permission Actions for this resource.



Add Permission

Select from permissions and resources used in this application. Enter search criteria to search for right permissions.
Customize resource or actions for selected permission.

Customize

Permission Class: oracle.security.jps.ResourcePermission

Resource Type: rtd_ils

Resource Name: Market_ILS

Permission Actions:

□ All

☑ open_service:write
□ decision_service:norma
☑ open_service:read
□ clear_statistics
☑ deploy_service

- Click Select.

You can also edit the Permissions and the Permission Actions, so long as you keep to the allowable Permission Actions and Action Qualifiers shown in Table 4–4.

Add One or More Grantees

You can add one or more Application Roles, Groups, Users.

For each addition, you can search and select from the available application roles, groups, and users.

After you have you finished creating the new application policy, the list of grantees that you included determines where the new application policy appears among the list of all security grants in the **obi** application stripe, as follows:

- If the grantees in your new application policy match the grantees of an existing security grant, as shown in the Principal column, the existing security grant showing those grantees will show the new application policy permissions for that grantee combination.
- If the grantees in your new application policy do not match the grantees of an existing security grant, as shown in the Principal column, a new Principal row

shows the new grantees and the permissions included in your new application policy.

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4.6 Creating an Application Policy Like Another Application Policy

The following is an overview of the process to create an application policy by copying from another application policy:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Policies.
3. Select and search for security grants in the Application Stripe **obi** (click the button beside the Permission box).
4. Select an Application Policy in the search results.
5. Click **Create Like...**
6. The Create Application Grant Like... page displays the permissions and grantees of the policy from which you want to copy. You can perform the following editing operations in the Create Application Grant Like... page:

   - Add, edit, and delete Permissions
   - Add and delete one or more Application Roles, Groups, Users

   The same considerations and recommendations apply as described in Section 4.7.4.5, "Creating a New Application Policy."

After you have you finished creating the new application grant, the list of grantees in the application grant determines where the new application grant appears among the list of all security grants in the **obi** application stripe, as follows:

- If the grantees in your new application grant match the grantees of an existing security grant, as shown in the Principal column, the existing security grant showing those grantees will show the new application grant permissions for that grantee combination.
- If the grantees in your new application grant do not match the grantees of an existing security grant, as shown in the Principal column, a new Principal row shows the new grantees and the permissions selected in the new application grant.

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4.7 Editing an Application Policy

The following is an overview of the process to edit an application policy:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Policies.
3. Select and search for security grants in the Application Stripe **obi** (click the button beside the Permission box).

4. Select an Application Policy in the search results.
5. Click **Edit...**
6. In the Edit Application Grant page:
   - Add, edit, or delete one or more Permissions
   - Add or delete one or more Application Roles, Groups, Users

   The same considerations and recommendations, both for the editing operations and for where to see the edited policy, apply as described in Section 4.7.4.6, "Creating an Application Policy Like Another Application Policy."

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.7.4.8 Deleting an Application Policy

The following is an overview of the process to delete an application policy:

1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, from either the server-level OracleRTD entry under Application Deployments, or the bifoundation_domain entry under WebLogic Domain, right-click and select Security, then Application Policies.
3. Select and search for security grants in the Application Stripe **obi** (click the button beside the Permission box).
4. Select an Application Policy in the search results.

   Note: The application policy that you select is identified by its grantee combination, as shown in the Principal column. The effect of deletion will be to remove the selected row, that is, the grantee combination together with its associated permissions, from the security grants list.

5. Click **Delete...**
6. Confirm that you want to delete the application policy.

For additional information and the detailed steps, see "Managing Policies with Fusion Middleware Control" in *Oracle Fusion Middleware Application Security Guide*.

4.8 Using SSL with Oracle RTD

For general information about SSL in Oracle Fusion Middleware, see the chapter "Configuring SSL in Oracle Fusion Middleware" in *Oracle Fusion Middleware Administrator's Guide*.

The instructions to enable SSL for Oracle RTD are as follows:

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

1. In the Oracle WebLogic Server Administration Console, after selecting Environments, then Servers, enable the default SSL port (9804) and Demo Trust for the Managed Server containing Oracle RTD.
2. On the client machine, create the directory `<RTD_HOME>/etc/ssl`, then copy the demo truststore file from the installed server-side location `<mw_home>/wlserver_10.3/server/lib/DemoTrust.jks` to `<RTD_HOME>/etc/ssl` on the client machine.
3. To make Decision Studio use the truststore file, edit `<MW_HOME>/Oracle_BI1/clients/rtd/eclipse/eclipse.ini` file as follows:

   Change the `-Djavax.net.ssl.trustStore` line to point to the truststore file.

   For example:

   ```
   -Djavax.net.ssl.trustStore=C:/OracleBI/RTD/etc/ssl/DemoTrust.jks
   ```
4. For CommandLineDeploy, execute `java -Djavax.net.ssl.trustStore=<DemoTrust.jks> -jar deploytool.jar -deploy -sslConnection true <ILS> <username> <password> <host> <port>`.

   For example:`java -Djavax.net.ssl.trustStore=C:/OracleBI/RTD/etc/ssl/DemoTrust.jks -jar deploytool.jar -deploy -sslConnection true "C:\OracleBI\RTD\examples\CrossSell" weblogic psw dadvmh0044 9804`
5. For Load Generator, in the script `<RTD_HOME>/scripts/sdexec.cmd`, uncomment the line:

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

   and replace the `TRUST_STORE_OPTS` value with the following one:

   ```
   -Djavax.net.ssl.trustStore=<dir-name>\DemoTrustStore.jks
   ```

   For example:

   ```
   -Djavax.net.ssl.trustStore=c:\rtd\etc\ssl\DemoTrust.jks
   ```
6. For Batch Console, execute `java -Djavax.net.ssl.trustStore="<DemoTrust.jks>" -jar batch-console.jar -url https://<machine_name>:<SSL_port>`

   For example, `java -Djavax.net.ssl.trustStore="c:\rtd\etc\ssl\DemoTrust.jks" -jar batch-console.jar -url https://localhost:9804`

4.9 Enabling SSO Authentication

This section provides some general guidelines for configuring single sign-on (SSO) authentication for Oracle RTD.

For general information about SSO in Oracle Fusion Middleware, see "Introduction to Single Sign-On in Oracle Fusion Middleware" in *Oracle Fusion Middleware Application Security Guide*.

Oracle recommends using Oracle Access Manager as an enterprise-level SSO authentication provider with Oracle Fusion Middleware 11*g*. For more information about configuring and managing Oracle Access Manager with Oracle Fusion Middleware, see "Configuring Single Sign-On with Oracle Access Manager 11g" and "Configuring Single Sign-On with Oracle Access Manager 10g" in *Oracle Fusion Middleware Application Security Guide*.

Specifically for Oracle RTD, the resource URLs to define, for both Authentication and Authorization Policies, are the following:

- Protected resources
  - /ui/*
- Public resources
  - /ui/images/*
  - /ui/do/logout
  - /ui/ping.jsp
  - /ui/logged_out.jsp

4.10 Topics of Interest in Other Guides

The following topics may be of interest to security administrators are covered in other guides. Table 4–8 lists these topics and the names of the guides where they can be found.

Table 4–8 Topics Covered in Other Guides

Topic	Guide Name
Installation	*Oracle Fusion Middleware Installation Guide for Oracle Business Intelligence*
Fusion Middleware Security Framework Managing Application Roles	*Oracle Fusion Middleware Application Security Guide*
Using Fusion Middleware Control	*Oracle Fusion Middleware Administrator's Guide*
Starting the Oracle WebLogic Server Administration Console	*Oracle Fusion Middleware Introduction to Oracle WebLogic Server*
Security for Oracle Business Intelligence	*Oracle Fusion Middleware Security Guide for Oracle Business Intelligence Enterprise Edition*

5
Configuring Data Access for Oracle Real-Time Decisions

This chapter describes how to configure the JDBC data sources which 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 information about supported versions for enterprise data sources, see the documents referred to in Section 1.4, "System Requirements and Certification."

This chapter contains the following topics:

- Section 5.1, "Creating Additional JDBC Data Sources in WebLogic"
- Section 5.2, "Testing a New Enterprise Data Source"

5.1 Creating Additional JDBC Data Sources in WebLogic

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

This section contains the following topics:

- Section 5.1.1, "Setting the Path to JDBC Jar Files for Your Data Source"
- Section 5.1.2, "Creating a Data Source in WebLogic"

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

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

1. If the enterprise data source you want to add is a Teradata data source, copy the files `terajdbc4.jar`, and `tdgssconfig.jar` to a directory of your choice, <*teradata_jar_files_directory*>. You can download these files from the Teradata Web site at `http://www.teradata.com`. Make sure that the JDBC driver files you download are compatible with the database version you are using.

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

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

3. Navigate the path:

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

4. Add the appropriate path or paths to **ClassPath**, including Windows or Linux/Unix separators between entries. Do not include spaces between your entries.

   - For Teradata

     `<teradata_jar_files_directory>\terajdbc4.jar`

     `<teradata_jar_files_directory>\tdgssconfig.jar`

5. Save.

6. Restart the WebLogic Managed Server.

5.1.2 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.

To create a data source in WebLogic:

1. Access the WebLogic Server Administration Console for the WebLogic domain in which Oracle RTD is deployed at the URL `http://weblogic_host:port/console`. At the login prompt, enter the administrator user name and password.

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.

   The Create a New JDBC Data Source window appears, showing the first page (JDBC Data Source Properties) of a sequence of pages that contain fields you must enter.

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**.

      For Oracle databases, select Oracle.

      For SQL Server databases, select MS SQL Server.

      For DB2 databases, select DB2.

      For other databases, select Other.

      Click **Next**.

   d. Set the **Database Driver**.

      For Oracle databases, select Oracle's Driver (Thin) for Service connections; Versions:9.0.1,9.2.0,10,11.

      For SQL Server databases, select Oracle's MS SQL Server Driver (Type 4) Versions:7.0, 2000, 2005, 2008.

      For DB2 databases, select Oracle's DB2 Driver (Type 4) Versions:7.X,8.X,9.X.

For other databases, select Other.

e. Click **Next**.

5. On the Transaction Options page:

   If you selected a non-XA JDBC driver for the Database Driver, the checkbox **Supports Global Transactions** appears. Deselect **Supports Global Transactions**, then click **Next**.

   If you selected an XA JDBC driver for the Database Driver, 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.

   b. For **Host Name**, enter the name of the computer hosting the database server. If you are adding an Oracle BI EE data source, enter the name of the computer hosting Oracle BI EE.

   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).

   d. For **Database User Name**, enter the name of the database run-time user. If you are adding an Oracle BI EE data source, enter the name of an Oracle BI EE user.

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

7. On the Test Database Connection page, for **Driver Class Name**, for Oracle, SQL Server and DB2 databases, accept all default settings. For other databases, 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):

   - **Teradata:** `com.teradata.jdbc.TeraDriver`
   - **Oracle BI EE:** `oracle.bi.jdbc.AnaJdbcDriver`

8. For URL, for Oracle, SQL Server and DB2 databases, accept all default settings. For other databases, enter the URL of the database to which you want to connect. The format of the URL varies by data source type:

   - **Teradata:** `jdbc:teradata://`*`server_name/db_name/param1,param2,...`*

     Note: If *`db_name`* is missing, the current login user's default database is used. For example, with default database **`RTD11G`**, `jdbc:teradata://64.181.232.117/TMODE=ANSI,CHARSET=ASCII` executes as `jdbc:teradata://64.181.232.117/`**`RTD11G`**`/TMODE=ANSI,CHARSET=ASCII`

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

     Note that the catalog name is required.

9. In the Properties field, for Oracle, SQL Server and DB2 databases, accept all default settings. For other databases, enter properties and their values required by the JDBC driver. The properties you need to provide vary by data source type:

   - For **Teradata**, enter the property `username=`*`db_user_name`*
   - For **Oracle BI EE**, there are no required properties. Leave the **Properties** field blank.

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

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

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

5.2 Testing a New Enterprise Data Source

After you add a new enterprise data source, follow the steps in this section to ensure the data source is configured properly. Before you begin, ensure that Oracle RTD is started.

To test a new enterprise data source:

1. Start Decision Studio by running `eclipse.exe` in *`RTD_HOME`*`\eclipse`. Then, create a new Inline Service, or open an existing Inline Service. See the Decision Studio Help for more information about how to do this.
2. Expand the **Service Metadata** folder, then right-click **Data Sources** and select **New SQL Data Source**.
3. For **Display Label**, enter a name for the data source you want to test, then click **OK**.
4. Click **Import**. Then, select the data source you want to test from the **JDBC Data Source** drop-down list. The list of Tables and Views is updated with tables and view names from that data source.
5. Select a particular table or view, then click **Finish** in the Import dialog box. The list of available columns appears in the Output table in the Data Source editor.
6. Write some basic code to ensure that the actual rows are retrieved from the tables at run time.

If you were not able to complete any of the preceding steps, check your data source configuration settings and try again.

6

Clustering and High Availability for Oracle Real-Time Decisions

This chapter introduces the concept of cluster deployment for Oracle RTD, and specifies where details of Oracle RTD clustering and high availability may be found.

Oracle RTD may be deployed into a cluster to achieve any of the following objectives:

- Increased processing power
- Increased memory, to accommodate more concurrent sessions
- Increased availability in the event of hardware failures

Full details of how to set up and use Oracle RTD clusters, both in general and in a high availability deployment, appear in the section "High Availability for Oracle Real-Time Decisions" in *Oracle Fusion Middleware High Availability Guide*.

7

Additional Configuration Settings and Starting Client Tools

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

This chapter contains the following topics:

- Section 7.1, "Decision Center Browser Configuration"
- Section 7.2, "Accessing Oracle Real-Time Decisions Client Tools"

7.1 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.

7.2 Accessing Oracle Real-Time Decisions Client Tools

Perform the steps in the following sections to start and access the Oracle RTD client-side tools, such as Decision Studio, Decision Center, and Load Generator. See Section 13.1, "Accessing the Oracle Real-Time Decisions MBeans" for information about the Oracle RTD MBeans.

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

This section contains the following topics:

- Section 7.2.1, "Accessing Decision Studio"
- Section 7.2.2, "Accessing Decision Center"
- Section 7.2.3, "Accessing Load Generator"

7.2.1 Accessing Decision Studio

In general, to access Decision Studio, go to the client computer where you installed the Oracle RTD client-side tools and run `*RTD_HOME*\eclipse\eclipse.exe`.

To run Decision Studio on a 64 bit machine, you must specify a 32 bit JDK on the command line, for example,

```
RTD_HOME\eclipse\eclipse.exe -vm "C:\jdk160_32bit\jre\bin\javaw"
```

7.2.2 Accessing Decision Center

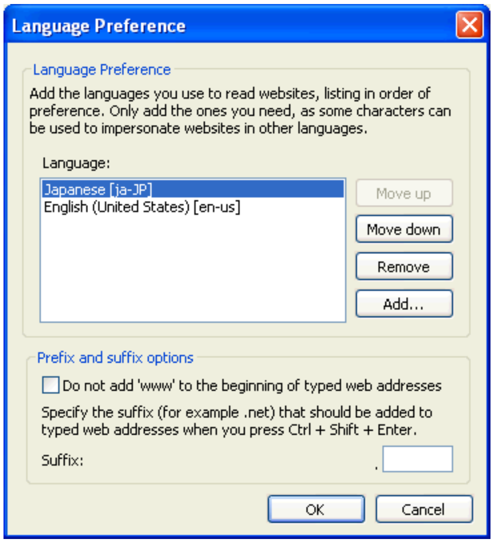
To access Decision Center from any computer, open a Web browser and go to `http://`*`server_name`*`:`*`port`*`/ui`.

The Oracle RTD application port is typically `9704`.

In the Sign In window, enter your User Name and Password, then click Sign In.

Note: To start up Decision Center in a language other than the default (English), you must set up the language setting in your browser.

For example, in Internet Explorer 7, navigate the path: `Tools > Internet Options > Languages`, then add or select your startup language, and move it to the top of the stack. In the following example the startup language is Japanese:



Add the languages you use to read websites, listing in order of preference. Only add the ones you need, as some characters can be used to impersonate websites in other languages.

Japanese [ja-JP]
English (United States) [en-us]

□ Do not add 'www' to the beginning of typed web addresses

Specify the suffix (for example .net) that should be added to typed web addresses when you press Ctrl + Shift + Enter.

Suffix: .

7.2.3 Accessing Load Generator

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

8

Production Deployment of Oracle Real-Time Decisions

This chapter introduces the concept of production deployment of Oracle RTD, and specifies where further production deployment details may be found.

After an Inline Service is tested and ready for production deployment, you will deploy it to one or more servers for production. For more information about the hardware requirement and the topologies in which Oracle RTD can be deployed, see the following topics in *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle Business Intelligence*:

- "Hardware Requirements for an Enterprise Deployment on Linux"
- "Overview of Enterprise Deployment Reference Topologies"

9 Command Line Deployment of Inline Services

In addition to deployment from Decision Studio, Inline Service deployment is also available from the command line. This type of Inline Service deployment can be performed either directly by a command in a command line window or from a script that calls the command. This chapter describes how to set up and use the Oracle RTD command line deployment tool.

This is useful in situations where several developers work on an Inline Service whose files are stored in a source control system. On a scheduled (or on-demand) basis, the Inline Service is retrieved from the source control system onto the production machine or onto another machine that can deploy the Inline Service into a production machine. You can then programmatically deploy the Inline Service to the production RTD server.

This chapter contains the following topics:

- Section 9.1, "Setting Up the Command Line Deployment Tool"
- Section 9.2, "Deploying the Inline Service at the Command Line"

9.1 Setting Up the Command Line Deployment Tool

As released, the command line deployment tool is located in the zip file, `rtd-deploytool-11.1.1.zip`, in the directory *RTD_HOME*`\client\CommandLineDeploy`.

First, unzip the file `rtd-deploytool-11.1.1.zip`. The file can be unzipped into any machine running Linux or Windows. The machine must have JDK installed. The JDK version used to run the command line deploy tool must be the same as the JDK version used by the RTD server.

In this chapter, the directory into which `rtd-deploytool-11.1.1.zip` is unzipped is referred to as *RTD_DEPLOYTOOL_DIR*. The unzipping creates the folder structure `OracleBI/RTD/deploytool` under *RTD_DEPLOYTOOL_DIR*. The main command line deployment tool, `deploytool.jar`, which performs the command line deployment, is located in the `RTDdeploytool` folder.

Note: To use the command line deployment tool, users must be associated with an application server role which permits deploying the Inline Service.

For example, if no permission assignments have been removed from the default roles after a Simple or Enterprise installation of Oracle RTD, then a user with the BIAuthor or BIAdministrator role is permitted to use the command line deployment tool.

9.2 Deploying the Inline Service at the Command Line

To deploy the Inline Service from the command line, perform the following steps:

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

1. Navigate to the directory containing the `deploytool.jar` file.

   ```
   cd RTD_DEPLOYTOOL_DIR/OracleBI/RTD/deploytool/
   ```

2. Run the following command:

   ```
   java [-Djavax.net.ssl.trustStore="<trust_store_location>"] -jar deploytool.jar
   -deploy [[<named_option>...] source [<positional_option> [...]]
   ```

   The parameters for `deploytool.jar` are as follows:

   - *`<named_option>...`* - one or more of the name-value pairs listed in Table 9–1.

Table 9–1 Command Line Deployment Named Options

Format	Description
`-server` *`address`*	Hostname or IP address for the RTD server. Default is localhost.
`-port` *`number`*	Port on which the RTD server listed. Default is 8080.
`-sslConnection` *`bool`*	Whether SSL is used for connection. Default is false.
`-deploymentState` *`name`*	Deployment state for this Inline Service.Default is Development
`-releaseLock` *`bool`*	Releasing the lock allows other users to edit the Inline Service in Decision Center or to deploy it from Decision Studio. Default is true.
`-terminateSessions` *`bool`*	Terminating active sessions in the Decision Server. Do not use this option when deploying to a production server unless necessary. Default is false.

- *`source`* - one of the following:
  - Directory: Full path to the Inline Service project folder which holds the Inline Service to be deployed.

  - Zip file: Full path to the zip file containing the Inline Service project folder which holds the Inline Service to be deployed.

    The zip file can contain only one Inline Service with up to one level parent folder.

  Paths that include spaces must be enclosed within double quotes, for example, `"C:\My Projects\CrossSell"`.

- *`<positional_option>[...]`* - positional parameter list in the format:

  `[username password [server[port[ssl[state[unlock[endsessions]]]]]]]`

  as listed in Table 9–2.

Table 9–2 Command Line Deployment Positional Options

Parameter	Description *(includes references to Table 9–1 Format entries)*
`username`	Name of a user logging in to the RTD server.
`password`	The user's password. For a blank password, specify "".
`server`	Positional `-server`
`port`	Positional `-port`
`ssl`	Positional `-sslConnection`
`state`	Positional `-deploymentState`
`unlock`	Positional `-releaseLock`
`endsessions`	Positional `-terminateSessions`

The command line deployment program deploys Oracle RTD Inline Services found in a source to a server from the command line.

The parameter `-deploy` is the first required argument.

The parameters *`source`*, *`username`*, and *`password`* are mandatory parameters, and are positional. If the values of the *`username`* and *`password`* parameters are not specified initially, users will be prompted to enter them.

If values are specified for a named option and its corresponding positional option, then the positional option value overrides the named option value.

Notes:

1. Help text listing and describing the parameters appears if the `jar` file is called without parameters, for example, `java -jar deploytool.jar`.
2. *`<trust_store_location>`* is the full path to the trust store file. Use the `-Djavax.net.ssl.trustStore` parameter only if `-sslConnection` (or the equivalent positional parameter `ssl`) is set to true.

Examples without named options

```
java -jar deploytool.jar -deploy "c:\my workspace\CrossSell"
scott brighton 192.168.0.15 8080 true Production false
```

```
java -jar deploytool.jar -deploy CrossSell sp34kc slater
```

```
java -jar deploytool.jar -deploy
c:\OracleBI\RTD\examples\CrossSell sdsu b21k7e false QA
```

Examples using named options

```
java -jar deploytool.jar -deploy -server 192.168.0.15 -port 8081
c:\OracleBI\RTD\examples\CrossSell sysman mi22ty
```

```
java -jar deploytool.jar -deploy -port 8081 -server 192.168.0.15
c:\OracleBI\RTD\examples\DC_Demo sonar chimney
```

```
java -jar deploytool.jar -deploy -sslConnection true
CrossSell.zip calzone twostep
```

```
java
-Djavax.net.ssl.trustStore="C:\OracleBI\RTDdeploytool\etc\ssl\sd
trust.store" -jar deploytool.jar -deploy -sslConnection true
-port 8443 "C:\OracleBI\RTD\examples\CrossSell" ssluser psword
```

10

Setting Up and Using Model Snapshots

The Oracle RTD Model Snapshot feature allows you to export data accumulated in Oracle RTD predictive models to external database tables. These results include counts of events, predictiveness values, and correlations. The data exported from the Oracle RTD models can then be analyzed using standard reporting and business intelligence products and techniques.

The data that Oracle RTD collects in its predictive models for a given Inline Service is attached to a Study. The association between an Inline Service and a Study is defined at deployment time.

The Model Snapshot functionality of Oracle RTD operates at the Study level and affects all the models defined in the Inline Service. Using Model Snaphots, you will be able to export the data contained in all the Models of an Inline Service for a given Study.

The data exported by the Model Snapshot feature allows you to replicate and extend the standard choice and choice group level "predictive model" reports provided by Oracle RTD Decision Center. Furthermore, when associated with customer data from a data warehouse, this exported data enables offline customer centric reporting of predictive insights collected and generated by Oracle RTD.

This section consists of the following topics:

- Section 10.1, "Overview of Setting Up and Using Model Snapshots"
- Section 10.2, "Model Snapshot Tables Schema"
- Section 10.3, "Configuring the Model Snapshot Tables"
- Section 10.4, "Populating and Clearing the Model Snapshot Tables"
- Section 10.5, "Creating Reports from the Model Snapshot Data"
- Section 10.6, "Handling Partitions"
- Section 10.7, "Tuning the Model Snapshot Process"

10.1 Overview of Setting Up and Using Model Snapshots

> **Note:** The description of how to set up and use the Model Snapshot functionality of Oracle RTD assumes that you have a running Inline Service with populated predictive models.

There are three main stages in the process of setting up and using model snapshots, as follows:

1. Configuring the model snapshot tables.
2. Populating the model snapshot tables from the learned data, and clearing the tables as required.
3. Creating reports from the model snapshot tables.

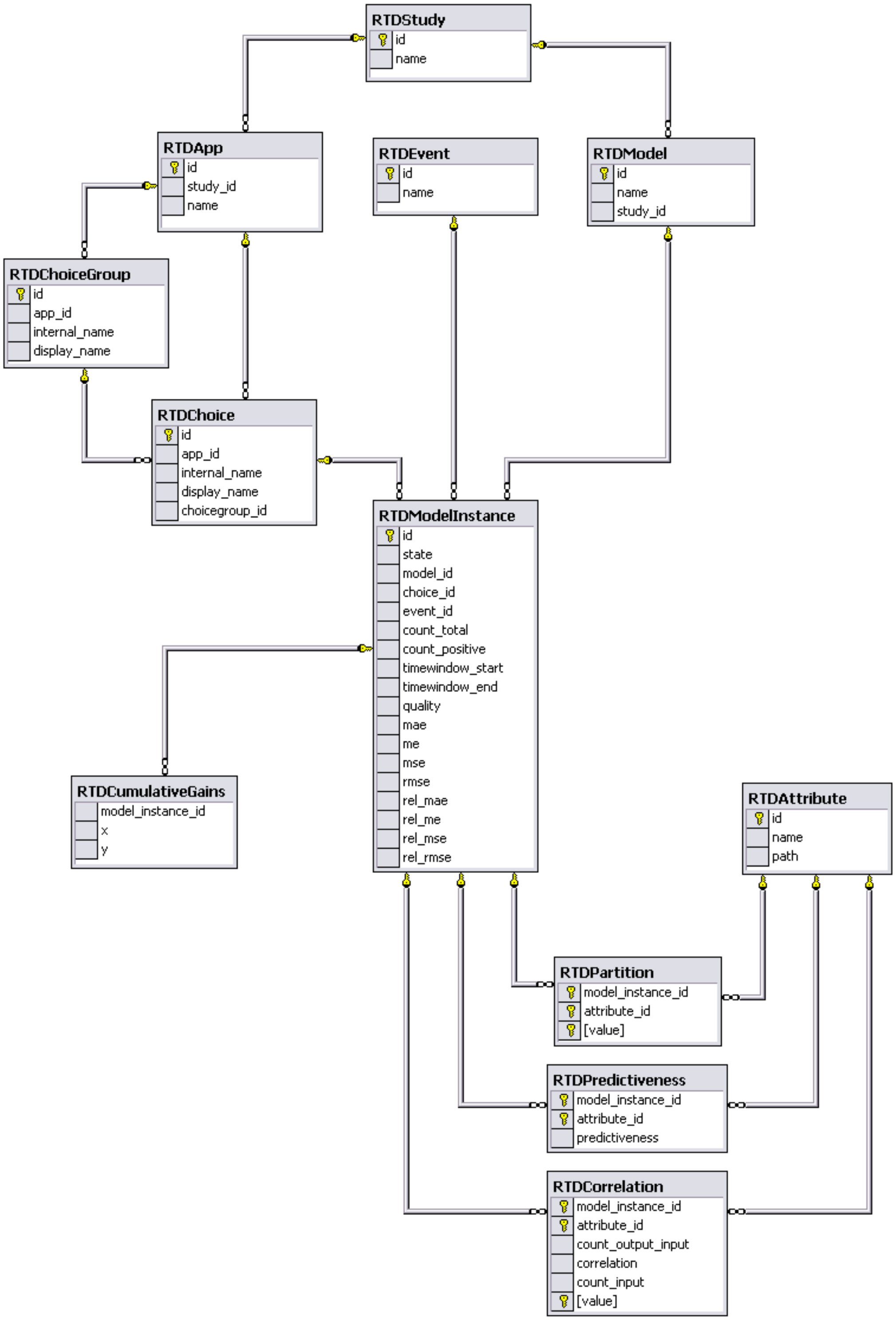
There are two parameters related to model snapshots that are for tuning purposes. For more information, see Section 10.7, "Tuning the Model Snapshot Process."

10.2 Model Snapshot Tables Schema

Oracle RTD exports its predictive model data using a multi-dimensional schema as described in the entity relationship model of Figure 10–1, which shows the model snapshot tables (except for RTDSnapshotSchemaVersion), their columns, and the inter-table relationships, as represented by the connector lines.

Each connector represents a one-to-many relationship, from the table at the top end of the connector to the table at the lower end. In the reverse direction, the relationship from the lower table to the top table is one-to-one.

Figure 10–1 Model Snapshot Tables Schema



The one-to-many relationships between the elements represented by the tables are as follows (each one-to-many relationship implies the corresponding one-to-one relationship in the reverse direction):

- Each Study can have one or more Applications, and one or more Models.

- Each Application can have one or more Choice Groups, and one or more Choices.
- Each Choice Group can have one or more Choices.
- Each Choice, Event, and Model can have one or more Model Instances.
- Each Model Instance can have one or more Cumulative Gains.

There are also three many-to-many relationships between Model Instances and Attributes, namely Partition, Predictiveness, and Correlation.

The table and column names appear in the following list.

- **RTDApp**

Column	Description
id	Primary key.
study_id	Foreign key to RTDStudy.
name	Application name.

- **RTDAttribute**

Column	Description
id	Primary key.
name	Attribute name.
path	Delimited attribute names showing how this attribute was reached from the root of the session.

- **RTDChoice**

Column	Description
id	Primary key.
app_id	Foreign key to RTDApp.
internal_name	Internal name for Choice.
display_name	Name displayed for Choice.
choicegroup_id	Foreign key to RTDChoiceGroup.

- **RTDChoiceGroup**

Column	Description
id	Primary key.
app_id	Foreign key to RTDApp.
internal_name	Internal name for Choice Group.
display_name	Name displayed for Choice Group.

- **RTDCorrelation**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
attribute_id	Foreign key to RTDAttribute.
count_output_input	Count of those in the population where the value of this output attribute was found given the input.
correlation	Value between -1 and 1. A positive correlation indicates the degree to which the input attribute's value is associated with the value of the output. A negative correlation indicates a negative association.
count_input	Count of those in the population where this value of this input attribute was found.
value	Value for the input attribute.

Note: The sum of over the **RTDCorrelation** column **count_input** for a particular model instance may be less than the related **RTDModelInstance** column **count_total**.

Also, the sum over the **RTDCorrelation** column **count_output_input** for a particular model instance may be less than the related **RTDModelInstance** column **count_positive**.

These results may occur in the following situations:

- Where rows were not stored in **RTDCorrelation** due to exceedingly low correlations, for example, too close to 0.
- Where the **RTDCorrelation** column **value** would have been null, meaning a value for the attribute was absent from the session. In these cases, no correlation is calculated.

- **RTDCumulativeGains**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
x	Cumulative gains curve data point.
y	Cumulative gains curve data point.

- **RTDEvent**

Column	Description
id	Primary key.
name	Event name.

- **RTDModel**

Column	Description
id	Primary key.
name	Model name.

Column	Description
study_id	Foreign key to RTDStudy.

- **RTDModelInstance**

Column	Description
id	Primary key.
state	Model state. Values can be: c: Completed. The model is completed, such as for previous time windows, and is no longer subject to change. It will not be rewritten unless you perform a total model snapshot, or delete the model snapshots and then perform an incremental model snapshot. d: Combined. The model is for the current and previous time window. s: Split. The model is for the current time window. w: Written. The model is currently being written. Results may be inconsistent.
model_id	Foreign key to RTDModel.
choice_id	Foreign key to RTDChoice.
event_id	Foreign key to RTDEvent.
count_total	Total number of base events.
count_positive	Size of the subset of the population where this non-base event was recorded.
time_window_start	Start of time window.
time_window_end	End of time window.
quality	Quality of the model, value vary from 0 to 1. Higher values are better, 0 means nothing was learned.
mae	Mean absolute error.
me	Mean error.
mse	Mean square error.
rmse	Root mean square error.
rel_mae	Relative mean absolute error.
rel_me	Relative mean error.
rel_mse	Relative mean square error.
rel_rmse	Relative root mean square error.

- **RTDPartition**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
attribute_id	Foreign key to RTDAttribute.
value	Attribute value for the partition.

- **RTDPredictiveness**

Column	Description
model_instance_id	Foreign key to RTDModelInstance.
attribute_id	Foreign key to RTDAttribute.
predictiveness	Explanatory score of an input attribute for a particular model instance. Values vary from 0 to 1, higher values are better.

- **RTDSnapshotSchemaVersion**

Column	Description
major	Major version.
minor	Minor version.

- **RTDStudy**

Column	Description
id	Primary key.
name	Study name.

10.3 Configuring the Model Snapshot Tables

> **Note:** The model snapshot tables are set up in the SDDB database during Simple and Enterprise installs of Oracle Real-Time Decisions. You may choose to store the model snapshot data in the SDDB database tables, but this is not recommended for a production system.

The main objectives of this stage are to create the model snapshot tables in a non-SDDS database, and to register them with the application server and the JMX MBeans.

> **Important:** The text values in the model snapshot tables must be case sensitive. If the default setting for your database is case insensitive, make sure that you override the setting when creating the model snapshot tables.

To configure the model snapshot tables:

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

1. Select the database where your model snapshot tables will be stored.
2. From the *RTD_HOME*`\scripts` directory, run the command that creates the model snapshot tables:

```
sdexec com.sigmadynamics.tools.SDDBTool.SDDBTool -f -i -I InitSnapshotDb.ctl
```

```
db_type db_host db_port db_name db_runtime_user db_admin_user db_admin_password
```

> **Note:** By replacing the **-i** parameter with **-u**, the operation changes from initialization of a new schema to an upgrade of an existing schema.

The following table describes the parameters for the `sdexec` script.

Parameter	Description
db_type	The database type. Select one of the following: oracle, sqlserver, db2.
db_host	The name of the computer hosting the database server.
db_port	The database port number.
db_name	The name of the database or, for Oracle Database, the SID.
db_runtime_user[1]	The user name of the run time user for the system.
db_admin_user	The name of a user that has rights on the database to create tables and stored procedures.
db_admin_password	The password of the administrative user.

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

3. Create a new Data Source in your application server, that references the database where your model snapshot tables are stored.

   For details of how to create a Data Source in an application server, see Chapter 5, "Configuring Data Access for Oracle Real-Time Decisions."

   As part of the operation of creating a new Data Source, you provide a new JNDI name, that is used in the steps following.

4. Go to the directory where you expanded the `RTD.ear` file during installation (*RTD_HOME*/`package/expanded`).

5. Open the `ls.war` archive, extract `web.xml`, then open `web.xml` for editing. Scroll to the bottom of the file. Copy the section for the definition of the resource reference of `SDDS_LS` and paste it after the existing section. In the copied section, replace the string `SDDS` with the JNDI name (*jndi_name*) that you provided in step 3.

   For example:

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

   Save the changes and close the file, then re-archive the file back in `ls.war`.

6. In Enterprise Manager, register the name of the new Data Source in the OracleRTD MBeans:

   - Navigate to MBeans > OracleRTD > SDClusterPropertyManager > Misc.

- For the **ModelSnapshotDSName** attribute, specify the name of the Data Source that you created for the model snapshots.

10.4 Populating and Clearing the Model Snapshot Tables

You can take snapshots of the model learning data at any time, even if not enough data has been accumulated for prediction purposes.

Each model is defined to have a time window, such as week, month, and quarter. This determines how much data to gather in the learning process, and also influences how much data to write to the model snapshot tables.

You can select the amount of data to write from the following options:

- All the model learning data for a Study
- All the model learning data for a Study for the current time window

You can also delete all model learning data for a Study from the model snapshot tables.

To populate or clear the model snapshot tables:

1. In Enterprise Manager, navigate to MBeans > OracleRTD > Learning Server > *your study name*.
2. Click the Operations tab.
3. Click the appropriate snapshot option:
   - **CompleteSnapshot**

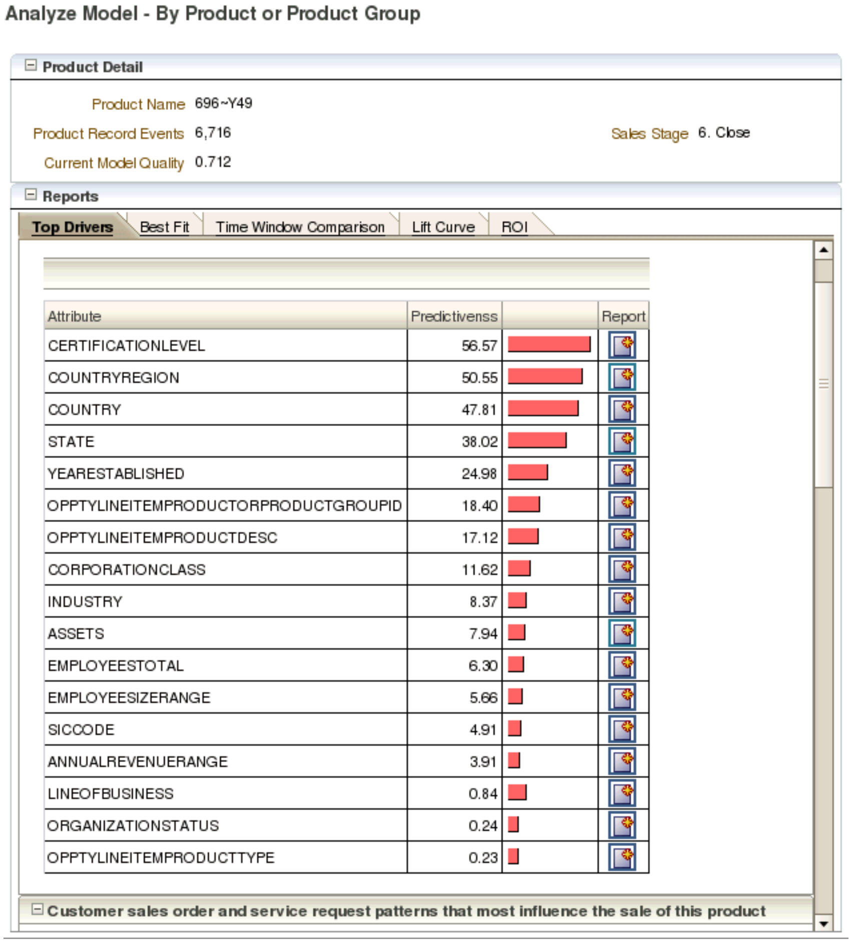
     Deletes all previously snapped data for the Study, and rewrites all Study data, up to the current time.
   - **IncrementalSnapshot**

     Deletes Study data for any incomplete time window, and rewrites the current time window's data, up to the current time.
   - **DeleteSnapshot**

     Deletes all snapped data for the Study.

10.5 Creating Reports from the Model Snapshot Data

You can create reports from the model snapshot tables, typically by using standard SQL Select commands, or by sending the data to business intelligence products which produce reports similar to the following product sales driver report:



Analyze Model - By Product or Product Group

⊟ Product Detail

Product Name 696~Y49

Product Record Events 6,716

Current Model Quality 0.712

Sales Stage 6. Close

⊟ Reports

Attribute	Predictivenss		Report
CERTIFICATIONLEVEL	56.57	🟥	
COUNTRYREGION	50.55	🟥	
COUNTRY	47.81	🟥	
STATE	38.02	🟥	
YEARESTABLISHED	24.98	🟥	
OPPTYLINEITEMPRODUCTORPRODUCTGROUPID	18.40	🟥	
OPPTYLINEITEMPRODUCTDESC	17.12	🟥	
CORPORATIONCLASS	11.62	🟥	
INDUSTRY	8.37	🟥	
ASSETS	7.94	🟥	
EMPLOYEESTOTAL	6.30	🟥	
EMPLOYEESIZERANGE	5.66	🟥	
SICCODE	4.91	🟥	
ANNUALREVENUERANGE	3.91	🟥	
LINEOFBUSINESS	0.84	🟥	
ORGANIZATIONSTATUS	0.24	🟥	
OPPTYLINEITEMPRODUCTTYPE	0.23	🟥	

⊟ Customer sales order and service request patterns that most influence the sale of this product

The rest of this section provides examples of scripts that extract information from several of the model snapshot tables. Each script example is followed by sample output. Notes are provided for some of the examples to help you to interpret the results.

The Inline Service used for these examples was a CrossSell application, and the data was generated by running the Oracle RTD Load Generator script to completion, simulating 400,000 user sessions.

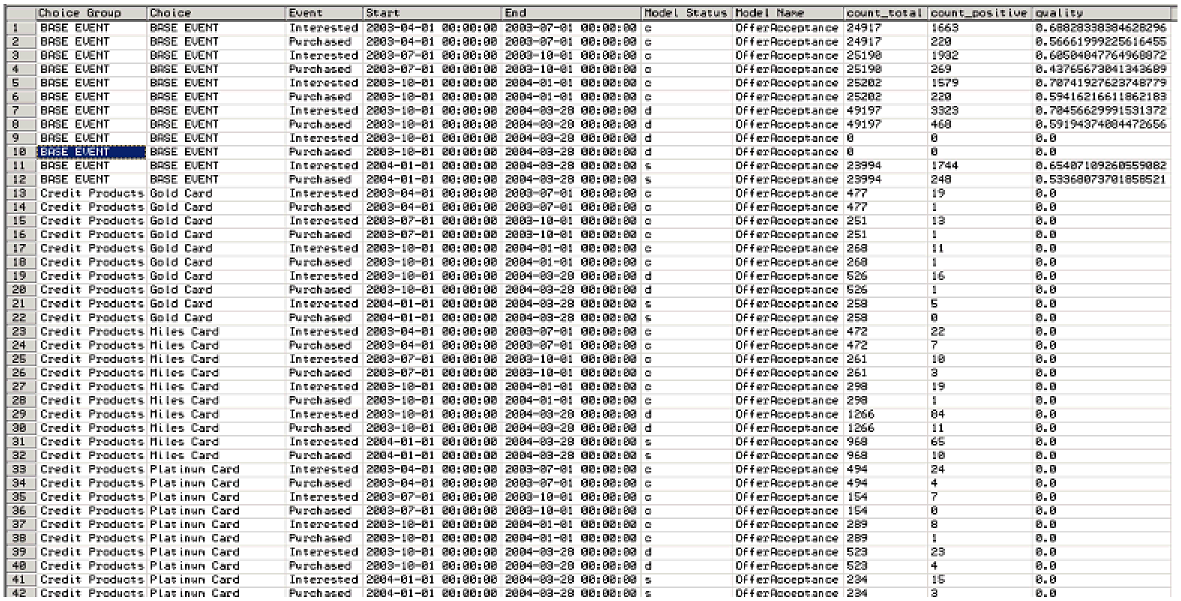
10.5.1 Counts by Choice Query

The following query gets the counts for every Choice, for all time windows:

```
select g.display_name     as 'Choice Group',
       c.display_name     as 'Choice',
       e.name             as 'Event',
      mi.timewindow_start as 'Start',
      mi.timewindow_end   as 'End',
      mi.state            as 'Model Status',
       m.name             as 'Model Name',
      mi.count_total,
      mi.count_positive,
      mi.quality
from RTDApp a
     inner join RTDStudy s           on s.id=a.study_id
     inner join RTDModel m           on m.study_id=s.id
     inner join RTDModelInstance mi  on mi.model_id=m.id
     inner join RTDEvent e          on mi.event_id=e.id
     inner join RTDChoice c         on c.id=mi.choice_id
     inner join RTDChoiceGroup g    on c.choicegroup_id=g.id
where a.name='CrossSell'
order by m.name,
         g.display_name,
         c.display_name,
       mi.timewindow_start
```

Figure 10–2 shows the results of the Counts by Choice query.

Figure 10–2 Counts by Choice Query Results



	Choice Group	Choice	Event	Start	End	Model Status	Model Name	count_total	count_positive	quality
1	BASE EVENT	BASE EVENT	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	24917	1663	0.68828338384628296
2	BASE EVENT	BASE EVENT	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	24917	220	0.56661999225616455
3	BASE EVENT	BASE EVENT	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	25190	1932	0.60504047764960872
4	BASE EVENT	BASE EVENT	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	25190	269	0.43765673041343689
5	BASE EVENT	BASE EVENT	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	25202	1579	0.70741927623748779
6	BASE EVENT	BASE EVENT	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	25202	220	0.59416216611862183
7	BASE EVENT	BASE EVENT	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	49197	3323	0.70456629991531372
8	BASE EVENT	BASE EVENT	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	49197	468	0.59194374084472656
9	BASE EVENT	BASE EVENT	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	0	0	0.0
10	BASE EVENT	BASE EVENT	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	0	0	0.0
11	BASE EVENT	BASE EVENT	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	23994	1744	0.65407109260559082
12	BASE EVENT	BASE EVENT	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	23994	248	0.53368073701858521
13	Credit Products	Gold Card	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	477	19	0.0
14	Credit Products	Gold Card	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	477	1	0.0
15	Credit Products	Gold Card	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	251	13	0.0
16	Credit Products	Gold Card	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	251	1	0.0
17	Credit Products	Gold Card	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	268	11	0.0
18	Credit Products	Gold Card	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	268	1	0.0
19	Credit Products	Gold Card	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	526	16	0.0
20	Credit Products	Gold Card	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	526	1	0.0
21	Credit Products	Gold Card	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	258	5	0.0
22	Credit Products	Gold Card	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	258	0	0.0
23	Credit Products	Miles Card	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	472	22	0.0
24	Credit Products	Miles Card	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	472	7	0.0
25	Credit Products	Miles Card	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	261	10	0.0
26	Credit Products	Miles Card	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	261	3	0.0
27	Credit Products	Miles Card	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	298	19	0.0
28	Credit Products	Miles Card	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	298	1	0.0
29	Credit Products	Miles Card	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	1266	84	0.0
30	Credit Products	Miles Card	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	1266	11	0.0
31	Credit Products	Miles Card	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	968	65	0.0
32	Credit Products	Miles Card	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	968	10	0.0
33	Credit Products	Platinum Card	Interested	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	494	24	0.0
34	Credit Products	Platinum Card	Purchased	2003-04-01 00:00:00	2003-07-01 00:00:00	c	OfferAcceptance	494	4	0.0
35	Credit Products	Platinum Card	Interested	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	154	7	0.0
36	Credit Products	Platinum Card	Purchased	2003-07-01 00:00:00	2003-10-01 00:00:00	c	OfferAcceptance	154	0	0.0
37	Credit Products	Platinum Card	Interested	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	289	8	0.0
38	Credit Products	Platinum Card	Purchased	2003-10-01 00:00:00	2004-01-01 00:00:00	c	OfferAcceptance	289	1	0.0
39	Credit Products	Platinum Card	Interested	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	523	23	0.0
40	Credit Products	Platinum Card	Purchased	2003-10-01 00:00:00	2004-03-28 00:00:00	d	OfferAcceptance	523	4	0.0
41	Credit Products	Platinum Card	Interested	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	234	15	0.0
42	Credit Products	Platinum Card	Purchased	2004-01-01 00:00:00	2004-03-28 00:00:00	s	OfferAcceptance	234	3	0.0

Notes on the Counts by Choice Query Results

1. In row 13, Choice=Gold Card, Event=Interested, count_total=477, count_positive=19, and quality=0.0. This shows that, out of 477 users that were presented with the Gold Card offer, 19 were interested. The counts are small and the model quality with respect to the Gold Card Choice and the Interested Event is 0.

   In row 14, Choice=Gold Card, Event=Purchased, count_total=477, count_positive=1, and quality=0.0. This shows that one user purchased this offer. The model quality with respect to the Gold Card Choice and the Purchased Event is 0.

   Both row 13 and row 14 apply to the period of time from April 1, 2003 to July 1, 2003.

2. In the columns Choice Group and Choice, the value BASE EVENT means "in general" or "overall."

   For example, in row 1, Choice=BASE EVENT, Event=Interested, count_total=24917, count_positive=1663, and quality is approximately 0.6882. This means that, in the period between the Start and End dates, a grand total of 24917 users were presented offers, and 1663 of these were interested. The overall model quality for this period of time was about 0.69.

   In row 2, Choice=BASE EVENT, Event=Purchased, count_total=24917, count_positive=220, and quality is approximately 0.5666. This means that, for the same period of time as for row 1, there were 220 Purchased events across all Choices, and the model quality was about 0.57.

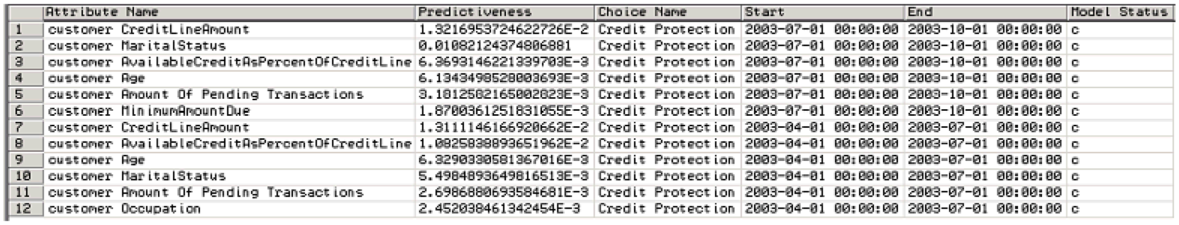
10.5.2 Top Six Predictive Attributes Query

The following query selects the top six predictive attributes, for each time window, for the Credit Protection Choice resulting in the Purchased Event.

```
select a.name                'Attribute Name',
       p.predictiveness      'Predictiveness',
       c.display_name        'Choice Name',
      mi.timewindow_start as 'Start',
      mi.timewindow_end   as 'End',
      mi.state            as 'Model Status'
from RTDApp app
     inner join RTDChoice c         on c.app_id=app.id
     inner join RTDStudy s          on s.id=app.study_id
     inner join RTDModel m          on m.study_id=s.id
     inner join RTDModelInstance mi on mi.model_id=m.id and mi.choice_id=c.id
     inner join RTDEvent e          on mi.event_id=e.id
     inner join RTDPredictiveness p on p.model_instance_id=mi.id
     inner join RTDAttribute a      on a.id=p.attribute_id
where app.name         = 'CrossSell'
  and   c.display_name = 'Credit Protection'
  and   e.name         = 'Purchased'
  and   m.name         = 'OfferAcceptance'
  and   7 > (select count(*)
             from RTDPredictiveness p2
             where p2.model_instance_id = p.model_instance_id
               and p2.predictiveness > p.predictiveness)
order by mi.timewindow_end desc,
          p.predictiveness desc
```

Figure 10–3 shows the results of the Top Six Predictive Attributes query.

Figure 10–3 Top Six Predictive Attributes Query Results



	Attribute Name	Predictiveness	Choice Name	Start	End	Model Status
1	customer CreditLineAmount	1.3216953724622726E-2	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
2	customer MaritalStatus	0.01082124374806881	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
3	customer AvailableCreditAsPercentOfCreditLine	6.3693146221339703E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
4	customer Age	6.1343498528003693E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
5	customer Amount Of Pending Transactions	3.1012502165002023E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
6	customer MinimumAmountDue	1.8700361251831055E-3	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
7	customer CreditLineAmount	1.3111146166920662E-2	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
8	customer AvailableCreditAsPercentOfCreditLine	1.0025838893651962E-2	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
9	customer Age	6.3290330581367016E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
10	customer MaritalStatus	5.4984893649816513E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
11	customer Amount Of Pending Transactions	2.6986880693584681E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
12	customer Occupation	2.452038461342454E-3	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c

10.5.3 Difference Between Expected and Actual Counts Query

The following query shows, for people of different marital statuses, the number who actually purchased credit protection and the number who were expected to do so. The report also shows the difference between the two values, and the importance of the correlation for the offer acceptance.

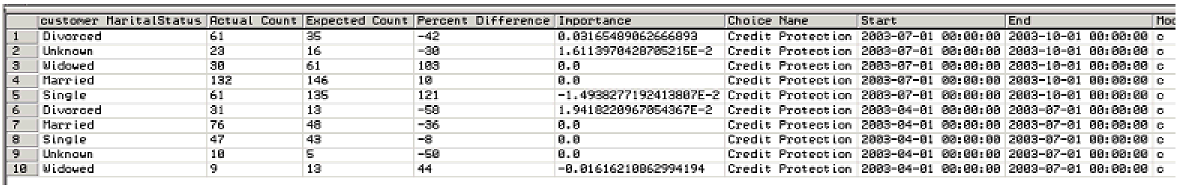
```
select cor.value               'customer MaritalStatus',
       cor.count_output_input  'Actual Count',
       mi.count_positive*cor.count_input/mi.count_total 'Expected Count',
  100* (  (mi.count_total*cor.count_output_input)
        / (mi.count_positive*cor.count_input) - 1) 'Percent Difference',
       cor.correlation         'Importance',
         c.display_name        'Choice Name',
        mi.timewindow_start as 'Start',
        mi.timewindow_end   as 'End',
        mi.state            as 'Model Status'
from RTDApp app
     inner join RTDChoice c         on c.app_id=app.id
     inner join RTDStudy s          on s.id=app.study_id
     inner join RTDModel m          on m.study_id=s.id
     inner join RTDModelInstance mi on mi.model_id=m.id and mi.choice_id=c.id
     inner join RTDEvent e          on mi.event_id=e.id
     inner join RTDCorrelation cor  on cor.model_instance_id=mi.id
     inner join RTDAttribute a      on a.id = cor.attribute_id
where app.name         = 'CrossSell'
    and c.display_name = 'Credit Protection'
    and e.name         = 'Purchased'
    and m.name         = 'OfferAcceptance'
    and a.name         = 'customer MaritalStatus'
order by mi.timewindow_end desc,
        cor.correlation desc
```

Notes on the Difference Between Expected and Actual Counts Query

1. The Actual Count, `cor.count_output_input`, is the actual number of people who purchased credit protection, for each marital status.
2. The Expected Count is a simple linear projection of the total count of each marital status, `cor.count_input`, to those that purchased credit protection, as expressed by `mi.count_positive/mi.count_total`.
3. The Percent Difference is 100 * (Actual Count - Expected Count) / Expected Count.

Figure 10–4 shows the results of the Difference Between Expected and Actual Counts query.

Figure 10–4 Difference Between Expected and Actual Counts Query Results



	customer MaritalStatus	Actual Count	Expected Count	Percent Difference	Importance	Choice Name	Start	End	Moc
1	Divorced	61	35	-42	0.03165489062666893	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
2	Unknown	23	16	-30	1.6113970428705215E-2	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
3	Widowed	30	61	103	0.0	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
4	Married	132	146	10	0.0	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
5	Single	61	135	121	-1.4938277192413807E-2	Credit Protection	2003-07-01 00:00:00	2003-10-01 00:00:00	c
6	Divorced	31	13	-58	1.9418220967054367E-2	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
7	Married	76	48	-36	0.0	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
8	Single	47	43	-8	0.0	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
9	Unknown	10	5	-50	0.0	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c
10	Widowed	9	13	44	-0.01616210062994194	Credit Protection	2003-04-01 00:00:00	2003-07-01 00:00:00	c

Notes on the Difference Between Expected and Actual Counts Query Results

1. There are two rows for each marital status, each corresponding to one of the two time periods, April 1, 2003 - July 1, 2003 and July 1, 2003 - October 1, 2003.

10.6 Handling Partitions

The RTDPartition table holds values for partitioning attributes. If a Model is not partitioned, the Model has one Model Instance per time window, and there are no associated rows in the RTDPartition table.

A Model that is split along one or more of its dimensions is a partitioned Model.

As an example, a Model M can be partitioned by two attributes, Marital Status and Favorite Beverage. If there are 3 values for Marital Status (Married, Single, Divorced) and 2 for Favorite Beverage (coffee, tea), then this model has 6 model instances.

In this case, each Model Instance has two associated RTDPartition rows. For example, the Model Instance for the combination (Marital Status=Married and Favorite Beverage=Coffee) would be associated with two RTDPartition rows, containing the following information:

- RTDPartition row 1: Attribute=Marital Status, Value=Married
- RTDPartition row 2: Attribute=Favorite Beverage, Value=Coffee

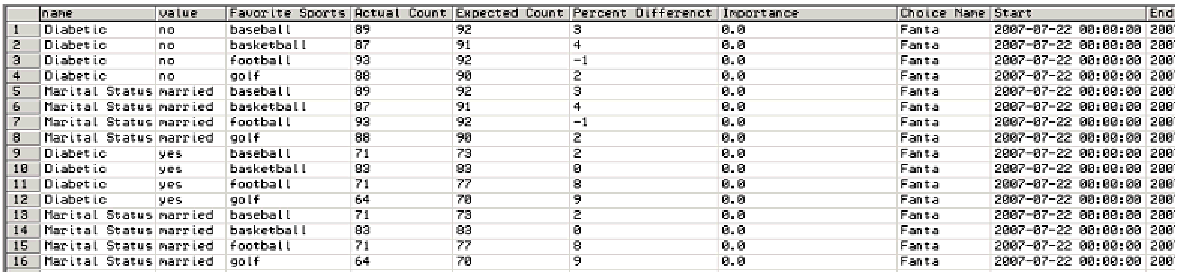
Whether or not a Model is partitioned can influence the results of queries on the model snapshot tables. To avoid repetitions in your results, include RTDPartition and RTDAttribute join conditions in your query.

The following example modifies and extends the Difference Between Expected and Actual Counts query to cover the case of a Model partitioned on two attributes, Diabetic, which has "yes" and "no" values, and Marital Status.

```
select a.name,
       p.value,
      subquery.*
from (select cor.value               'Favorite Sports',
             cor.count_output_input  'Actual Count',
             mi.count_positive*cor.count_input/mi.count_total 'Expected Count',
  100* (  (mi.count_total*cor.count_output_input)
        / (mi.count_positive*cor.count_input) - 1) 'Percent Difference',
       cor.correlation         'Importance',
         c.display_name        'Choice Name',
        mi.timewindow_start as 'Start',
        mi.timewindow_end   as 'End',
        mi.state            as 'Model Status'
        mi.id                   model_instance_id
  from RTDApp app
       inner join RTDChoice c         on c.app_id=app.id
       inner join RTDStudy s          on s.id=app.study_id
       inner join RTDModel m          on m.study_id=s.id
       inner join RTDModelInstance mi on mi.model_id=m.id and mi.choice_id=c.id
       inner join RTDEvent e          on mi.event_id=e.id
       inner join RTDCorrelation cor  on cor.model_instance_id=mi.id
       inner join RTDAttribute a      on a.id = cor.attribute_id
  where app.name       = 'HighlyPartitionedDataset'
    and c.display_name = 'Fanta'
    and e.name = 'loved'
    and m.name = 'SatisfactionModel'
    and a.name = 'Favorite Sports') as subquery
inner join RTDPartition p on subquery.model_instance_id = p.model_instance_id
inner join RTDAttribute a on p.attribute_id = a.id
order by subquery.[End] desc,
         subquery.model_instance_id,
         a.name,
         p.value
         subquery.[Importance] desc
```

Figure 10–5 shows the results of the Partitioned Expected and Actual Counts query.

Figure 10–5 Partitioned Expected and Actual Counts Query Results



	name	value	Favorite Sports	Actual Count	Expected Count	Percent Differenct	Importance	Choice Name	Start	End
1	Diabetic	no	baseball	89	92	3	0.0	Fanta	2007-07-22 00:00:00	200
2	Diabetic	no	basketball	87	91	4	0.0	Fanta	2007-07-22 00:00:00	200
3	Diabetic	no	football	93	92	-1	0.0	Fanta	2007-07-22 00:00:00	200
4	Diabetic	no	golf	88	90	2	0.0	Fanta	2007-07-22 00:00:00	200
5	Marital Status	married	baseball	89	92	3	0.0	Fanta	2007-07-22 00:00:00	200
6	Marital Status	married	basketball	87	91	4	0.0	Fanta	2007-07-22 00:00:00	200
7	Marital Status	married	football	93	92	-1	0.0	Fanta	2007-07-22 00:00:00	200
8	Marital Status	married	golf	88	90	2	0.0	Fanta	2007-07-22 00:00:00	200
9	Diabetic	yes	baseball	71	73	2	0.0	Fanta	2007-07-22 00:00:00	200
10	Diabetic	yes	basketball	83	83	0	0.0	Fanta	2007-07-22 00:00:00	200
11	Diabetic	yes	football	71	77	8	0.0	Fanta	2007-07-22 00:00:00	200
12	Diabetic	yes	golf	64	70	9	0.0	Fanta	2007-07-22 00:00:00	200
13	Marital Status	married	baseball	71	73	2	0.0	Fanta	2007-07-22 00:00:00	200
14	Marital Status	married	basketball	83	83	0	0.0	Fanta	2007-07-22 00:00:00	200
15	Marital Status	married	football	71	77	8	0.0	Fanta	2007-07-22 00:00:00	200
16	Marital Status	married	golf	64	70	9	0.0	Fanta	2007-07-22 00:00:00	200

10.7 Tuning the Model Snapshot Process

You can tune the model snapshot process through the following parameters, available as JMX MBean attributes:

- **ModelSnapshotMinAbsCorrelation** controls whether to snapshot all correlation rows or to set a minimum correlation value for snapshots.
- **ModelSnapshotNumberOfBins** controls the number of bins for model snapshots.

For more information, see Section 13.3.2, "About OracleRTD > SDClusterPropertyManager > Misc."

11

Performance Monitoring

Oracle Real-Time Decisions includes a robust performance monitoring system for observing the behavior of Inline Services. Performance Monitoring parameters are set, and a snapshot view of some of the common counters can be observed, through Fusion Middleware Control. A chronological view can be obtained by enabling the performance monitor. Once enabled, a comma-separated value (CSV) file is produced that can be used to observe behavior over time.

Caution: This file grows without limit, and should be enabled only for active troubleshooting.

This section contains the following topics:

- Section 11.1, "Setting Performance Monitoring Parameters"
- Section 11.2, "Viewing Common Performance Monitoring Snapshot Values"
- Section 11.3, "CSV File Contents"
- Section 11.4, "XLS File Contents"
- Section 11.5, "Switching Off Authentication and Authorization"

11.1 Setting Performance Monitoring Parameters

The performance monitoring parameters are set using the SDManagementCluster > Members > Properties > PerformanceMonitoring MBean. You can access this MBean using Fusion Middleware Control; see Chapter 13, "Managing Oracle Real-Time Decisions" for more information.

Table 11–1 describes the properties governing performance monitoring.

Table 11–1 Performance Monitoring Properties

Property Name	Description
DSPerfCounterEnabled	Enables the writing of DS performance counters. This property should not be enabled indefinitely, because the file grows without limit.
DSPerfCounterAppend	If true, performance data is appended to an existing file, if any, otherwise any existing file is overwritten when the server restarts.

Table 11–1 (Cont.) Performance Monitoring Properties

Property Name	Description
DSPerfCounterLogFile	The tab-separated CSV file into which DS performance counts are periodically appended. If MS Excel is available, `ds_perf.xls`, supplied in the `etc` directory of the installation, provides a convenient view. See the first row of `ds_perf.xls` for instructions on linking `ds_perf.xls` to `ds_perf.csv` as a datasource.
DSPerfCounterLogInterval	The update interval in milliseconds for DS performance counts.

11.2 Viewing Common Performance Monitoring Snapshot Values

A snapshot of some of the performance counters is available for viewing through the SDManagementCluster > Members > Decision Service MBean. Press the F5 key to refresh the values.

Performance monitoring does not have to be enabled to use this view.

11.3 CSV File Contents

This section describes the fields of the CSV file containing performance counters.

Table 11–2 Fields of CSV File With Performance Counters

Field Name	Description
Date/Time	The time of day at which the current row of counters was appended to the file. Millisecond precision is available to facilitate correlations with messages in the server's log file.
Peak Requests Running	The largest number of requests that have been running at the same time since the server was started.
Max Requests Running	The largest number of requests that have been running at the same time during the current logging interval.
Requests Running	The number of Inline Service requests that are currently running. This value will always be less than or equal to the Max Allowable Running Requests value.
Requests Finished	The total number of Inline Service requests that have been finished.
Requests Seen, Total	The total number of Inline Service requests for this server.
Requests In System	The current number of Inline Service requests being processed by this server. The number includes those waiting to run, and those already running.
Timed Out Requests, Total	The total number of requests that have failed to finish running before their guaranteed service level timeout, as specified by cluster-wide property SDManagementCluster > Properties > Misc > IntegrationPointGuaranteedRequestTimeout. This count includes all timed-out requests since the server was started. If this number is growing but the number of queued requests is not growing, this is an indication that the Inline Service logic handling the request is too slow to satisfy the response time guarantee, even on an idle system. One or more Integration Point requests must be optimized, or the response time guarantee must be increased.

Table 11–2 (Cont.) Fields of CSV File With Performance Counters

Field Name	Description
Timed Out Requests	The number of requests that failed to finish running before their guaranteed service level.
Timed Out While Running, Total	The total number of requests, observed since the server started, to have started running and not finish within their response time guarantee. The server's processing power consumed by these requests is largely wasted, because the clients will ignore their late responses. When the system is very busy, it sometimes times out requests that are still waiting to run, thus avoiding wasting resources on them.
Timed Out While Running	The number of requests, observed during the current logging interval started, to have started running and not finish within their response time guarantee. The server's processing power consumed by these requests is largely wasted, because the clients will ignore their late responses. When the system is very busy, it sometimes times out requests that are still waiting to run, thus avoiding wasting resources on them.
Timed Out Requests Still Running	The number of requests that have started running, timed out, and are still running. A non-zero value could be an indication of a programming problem in one or more Integration Points.
Request Run Time, Average (ms)	The average time, in milliseconds, during the current logging interval that requests ran. Excludes wait time, if any.
Request Run Time, Max (ms)	The largest amount of time, in milliseconds, during the current logging interval, that any single request ran. Excludes wait time, if any.
Run Times < [0.1 GRT]	The number of requests that finished running during the current logging interval and ran less than 10% of the configured guaranteed response time. There are nine similarly formatted columns, showing the run time distribution for 0.10, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, and 2.0 times the guaranteed response time.
Run Times < N and >= M	The number of requests that finished running during the current logging interval and ran less than N milliseconds and greater than or equal to M milliseconds.
Run Times >= [2.0 GRT]	The number of requests that finished running during the current logging interval and ran two or more times the configured guaranteed response time.
Request Wait Time, Average (ms)	The average time, in milliseconds, that requests waited on the request queue prior to running or timing out.
Request Wait Time, Max (ms)	The largest amount of time, in milliseconds, during the current logging interval, that any single request waited on the request queue. Includes only those requests that finished running, or timed out before running, during the current logging interval.
Wait Times < [0.1 GRT]	The number of requests that finished running during the current logging interval, and were placed on the request queue before running, but waited there less than 10% of the configured guaranteed response time. There are nine similarly formatted columns, showing the wait time distribution for 0.10, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, and 2.0 times the guaranteed response time.

Table 11–2 (Cont.) Fields of CSV File With Performance Counters

Field Name	Description
Wait Times < N and >= M	The number of requests that finished running during the current logging interval and waited on the request queue less than N milliseconds and greater than or equal to M milliseconds before running.
Wait Times >= [2.0 GRT]	The number of requests that finished running during the current logging interval and waited two or more times the configured guaranteed response time before timing out.
Sessions, Current	The number of Decision Server sessions still open in this server.
Sessions, Total	The total number of Decision Server sessions created by this server.
Stale Sessions Closed Asynchronously	The total number of Decision Server sessions that have been closed by kernel jobs, instead of by request threads. This is usually unimportant. In a busy system, most stale sessions are closed by request threads and the kernel jobs are engaged only as the system winds down. It could be of interest to someone observing a lot of kernel-job activity (see **Kernel Jobs Running, Current**).
Stale Sessions Closed by Requests	The total number of Decision Server sessions that have timed out and been closed by request threads. Most sessions will be closed this way, especially on a busy server. After processing an Inline Service request, the calling thread will be asked to close at most one stale session before returning to the caller.
Requests Forwarded, Current	The total number of Inline Service requests that have been forwarded from this server to other servers, and for which no acknowledgment has yet been received to indicate that the request has been processed by the forwarded-to server.
Requests Forwarded, Peak	Largest number of Inline Service requests forwarded.
Requests Forwarded, Total	Total number of Inline Service requests forwarded.
Received Requests Forwarded, Current	The total number of Inline Service requests that were forwarded from other servers to this server, and which have not yet been completely processed by this server.
Received Requests Forwarded, Peak	Largest number of received Inline Service requests forwarded.
Received Requests Forwarded, Total	Total number of received Inline Service requests forwarded.
Remote Session Keys, Current	The current number of session keys that this server knows reference sessions hosted by other servers. If a request arrives with one of these keys, it will be forwarded to the other server.
Remote Session Keys, Total	The total number of times that session keys were registered in this server for sessions hosted by other servers. This is an aggregation of "Remote Sessions Keys, Current".
Kernel Jobs Running, Current	The number of maintenance activities currently running in the server. Maintenance activities include model maintenance, session timing, and timed-out request processing.
Kernel Jobs Running, Peak	The largest number of maintenance activities that have run at the same time in this server. This value will always be less than or equal to the cluster-wide property SDManagement-Cluster > Properties > Misc > WorkerThreadPoolSize, or the server-specific property, SDManagement-Cluster > Members > Properties > Misc > WorkerThreadPoolSize.

Table 11–2 (Cont.) Fields of CSV File With Performance Counters

Field Name	Description
Snapshot Period (ms)	The period of time, in milliseconds, over which the server collected data before logging this row of counters.

11.4 XLS File Contents

This section describes the contents of the Microsoft Excel file, `ds_perf.xls`, included in the `etc` directory of the installation.

At the top, cell B1 contains a comment describing how to link `ds_perf.xls` to the tab-separated counter file as a datasource:

"To specify path to the ds_perf.csv file, place cursor in cell B2 and select "Import External Data" > "Edit Text Import" from the "Data" menu, and navigate to your {$install_directory}\log\ folder and select the ds_perf.csv file. Use default parsing settings when prompted. Data will then be automatically refreshed every 3 minutes. To change interval and other settings, select from the "Data" menu the selection "Import External Data" > "Data Range Properties""

In row 2 are the headers containing the names of each counter. All of the headers from the CSV file appear here, with values below them.

The following columns appear after the values from the CSV file, with formulas showing values calculated from the CSV values:

- **Gross Throughput (req/sec):** The average rate of requests finishing during the current logging interval, in requests per second. The formula is:

  ```
  RequestsFinished / SnapshotPeriod * 1000
  ```

- **Net Throughput (req/sec):** The average rate of requests finishing during the current logging interval, excluding requests that timed out. The formula is:

  ```
  (RequestsFinished - Timeouts) / SnapshotPeriod * 1000
  ```

11.5 Switching Off Authentication and Authorization

By default, Oracle RTD has authentication and authorization switched on. In order to improve performance in Decision Server, you can switch off authorization and authentication.

For Oracle RTD Decision Server authorization to be switched off, perform the following steps:

1. Log in to Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the Target Navigation Pane, under Application Deployments, right-click the OracleRTD server entry, then select System MBean Browser.
3. In the System MBean Browser, scroll down to the Application Defined MBeans, select OracleRTD and then the server name where Oracle RTD is deployed.
4. Select SDClusterPropertyManager, then Cluster.
5. Set RequireIntegrationPointAuthorization to false.

You can switch off Decision Server authentication on a web service. For details, see "Web Service Security" in *Oracle Fusion Middleware Platform Developer's Guide for Oracle Real-Time Decisions*.

12

Backup and Recovery of Oracle Real-Time Decisions

This chapter introduces the concept of backup and recovery for Oracle RTD, and specifies where details of backup and recovery for Oracle RTD may be found.

Backup and recovery refers to the various strategies and procedures involved in guarding against hardware failures and data loss, and reconstructing data should loss occur.

Backup and recovery for Oracle Real-Time Decisions are fully described in in *Oracle Fusion Middleware Administrator's Guide*. For more information, see the section "Backup and Recovery Recommendations for Oracle Real-Time Decisions" in *Oracle Fusion Middleware Administrator's Guide*.

13

Managing Oracle Real-Time Decisions

Oracle RTD uses the J2EE industry standard Java Management Extensions (JMX) to configure and monitor the operation of Oracle RTD. This chapter describes how to manage Oracle RTD through the Oracle RTD JMX MBeans.

Oracle RTD systems deployed into WebLogic are managed using the Fusion Middleware Control MBean Browsers in Enterprise Manager.

> **Note:** Some of the management properties can also be set as system properties. For details, see Section 13.9, "System Properties."

JMX MBeans manage various aspects of Oracle RTD, including logging and Inline Service configuration. They can also be used to assign Oracle RTD specific permissions to security roles and users. Security roles and users are both managed by the J2EE container.

Oracle RTD is comprised of three Services:

- **Decision Center Service:** Supports the deployment of Inline Services by Decision Studio. It also provides a web interface, Decision Center, for displaying the structure and decisioning history of Inline Services.
- **Decision Service:** Runs Inline Services and integrates to enterprise operational processes.
- **Learning Service:** Maintains analytic, self-learning models that underlie Inline Services.

Deployments of Oracle RTD are often done across multiple servers as well as in clusters to enhance performance in high transaction environments. A relational database is used by each of these Services for retention of code, transactional data and configurations.

> **Note:** This chapter deals mainly with the management of Oracle RTD through JMX MBeans in Enterprise Manager. Administrators can also perform certain Inline Service related administrative tasks using web service calls in their own application utilities. For more information, see the section "Administration Web Service" in *Oracle Fusion Middleware Platform Developer's Guide for Oracle Real-Time Decisions*.

This chapter contains the following topics:

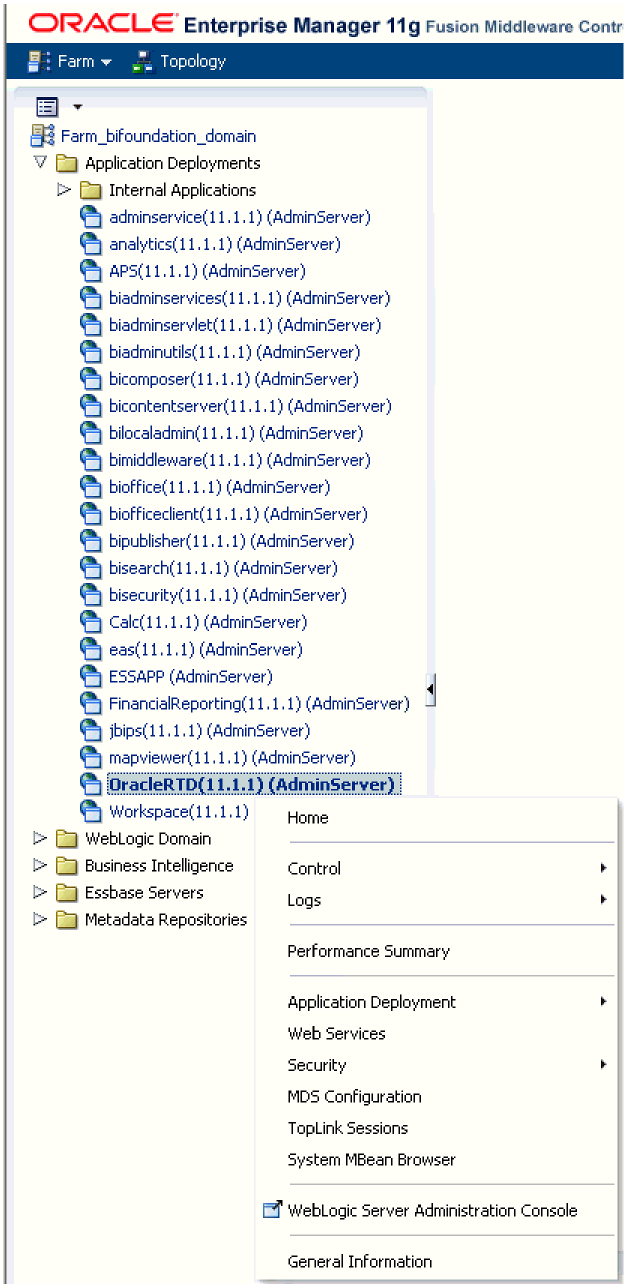
- Section 13.2, "About JMX MBean Operations and Attributes"
- Section 13.3, "MBeans for Oracle Real-Time Decisions Cluster-Level Management"
- Section 13.4, "MBeans for Oracle Real-Time Decisions Member-Level Management"
- Section 13.5, "MBeans for Managing Inline Services"
- Section 13.6, "MBeans for Deployment States"
- Section 13.7, "MBeans for Managing Learning Services"
- Section 13.8, "Post-Deployment Management of Inline Services"
- Section 13.9, "System Properties"

13.1 Accessing the Oracle Real-Time Decisions MBeans

Oracle RTD under WebLogic is a component of Oracle Fusion Middleware. You access the Oracle RTD JMX MBeans through the System MBean Browser that exists within Oracle Enterprise Manager Fusion Middleware Control.

To access the System MBean Browser for Oracle RTD, you must perform the following steps:

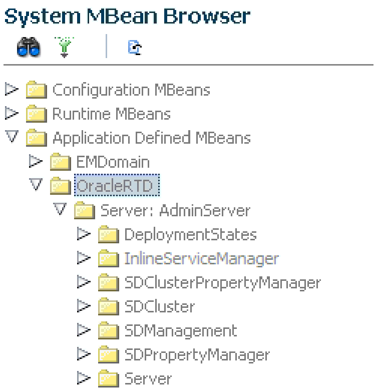
1. Log into Fusion Middleware Control, as described in Section 2.1.1, "Logging into Fusion Middleware Control."
2. In the left-hand Target Navigation Pane, after selecting Application Deployments, then Internal Applications, right-click the Oracle RTD node (as shown in the following image) or the Oracle RTD node within the cluster or server that contains Oracle RTD.


ORACLE Enterprise Manager 11g Fusion Middleware Contr
Farm
Topology
Farm_bifoundation_domain
Application Deployments
Internal Applications
adminservice(11.1.1) (AdminServer)
analytics(11.1.1) (AdminServer)
APS(11.1.1) (AdminServer)
biadminservices(11.1.1) (AdminServer)
biadminservlet(11.1.1) (AdminServer)
biadminutils(11.1.1) (AdminServer)
bicomposer(11.1.1) (AdminServer)
bicontentserver(11.1.1) (AdminServer)
bilocaladmin(11.1.1) (AdminServer)
bimiddleware(11.1.1) (AdminServer)
bioffice(11.1.1) (AdminServer)
bioffficeclient(11.1.1) (AdminServer)
bipublisher(11.1.1) (AdminServer)
bisearch(11.1.1) (AdminServer)
bisecurity(11.1.1) (AdminServer)
Calc(11.1.1) (AdminServer)
eas(11.1.1) (AdminServer)
ESSAPP (AdminServer)
FinancialReporting(11.1.1) (AdminServer)
jbips(11.1.1) (AdminServer)
mapviewer(11.1.1) (AdminServer)
OracleRTD(11.1.1) (AdminServer)
Workspace(11.1.1)
WebLogic Domain
Business Intelligence
Essbase Servers
Metadata Repositories
Home
Control
Logs
Performance Summary
Application Deployment
Web Services
Security
MDS Configuration
TopLink Sessions
System MBean Browser
WebLogic Server Administration Console
General Information

3. Select System MBean Browser.

4. To access the Oracle RTD MBeans, scroll down the System MBean Browser left-hand pane, and select Application Defined MBeans, then Oracle RTD, then the server where Oracle RTD is deployed.

The following example shows the MBean grouping categories for Oracle RTD:



For more information about MBeans and MBean Browsers, see *Oracle Fusion Middleware Administrator's Guide*.

13.2 About JMX MBean Operations and Attributes

Oracle RTD MBeans can be accessed through the OracleRTD folder in the System MBean Browser. These MBeans can be used to manage various aspects of Oracle RTD. Each MBean consists of attributes and operations that can be used for informational and administration purposes. The attributes and operations described in this chapter are specific to Oracle RTD.

13.3 MBeans for Oracle Real-Time Decisions Cluster-Level Management

Management at the cluster level is for items that impact the entire cluster of servers. Note that if you have only one server, there is still cluster-level management.

This section contains the following topics:

- Section 13.3.1, "About OracleRTD > SDManagement > SDClusterPropertyManager"
- Section 13.3.2, "About OracleRTD > SDClusterPropertyManager > Misc"
- Section 13.3.3, "About OracleRTD > SDClusterPropertyManager > Cluster"
- Section 13.3.4, "About OracleRTD > SDClusterPropertyManager > Deployment"
- Section 13.3.5, "About OracleRTD > SDCluster > SDManagement"

13.3.1 About OracleRTD > SDManagement > SDClusterPropertyManager

The SDManagement > SDClusterPropertyManager MBean has the following attributes:

Attribute	**Description**
Cluster	Cluster configuration.
Deployment	Configuration for deployment.
Misc	Miscellaneous properties.

The SDManagement > SDClusterPropertyManager MBean has the following operation:

java.lang.Void restoreDefault()

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

13.3.2 About OracleRTD > SDClusterPropertyManager > Misc

The SDClusterPropertyManager > Misc MBean has the following attributes:

Attribute	Description
ArchivedModelCacheTimeToLive	Maximum time in seconds an archived model is preserved in memory. The cache of archived models is used by Discovery Explorer.
ArchivedModelCatalogRefreshInterval	Refresh interval in seconds for a catalog of archived models. The catalog is used by Discovery Explorer.
AutoFlushTimeout	Interval in seconds controlling auto flush of database write buffers. Fractional values are supported.
DBOperationLogThresholdMilliSec	All database operations that take longer than the specified threshold are logged.
DCOperationLogThresholdMilliSec	All decision center requests that take longer than the specified threshold are logged.
DSManagesSessionAffinity	Decision Service manages session affinity. When set to true, the decision service maintains a map of active session keys and, if necessary, will forward Integration Point requests to the cluster host owning the key's session. Should be disabled in single-host installations and in installations where session affinity is perfectly managed by the application server or external load balancer.
DSSessionIdleTimeoutMillisec	Decision Service session idle timeout in milliseconds.
DSStrictSessionAffinityConcurrency	How many concurrent DB requests can be outstanding, per Oracle RTD instance, to manage the strict Decision Service session directory. Keep this low for SQL Server to avoid DB-centric deadlocks. 0 means unlimited.
DatabaseComponentCloseTimeoutSeconds	How long to wait for the database provider to close an individual component (for example, each BatchUpdater) before abandoning the effort.
DatabaseShutdownTimeoutSeconds	How long to wait for the database provider to shutdown all its components before abandoning the effort.
DecisionServiceAddress	Must be set for Decision Center to be able to test integration point requests from its Interactive Integration Map when Decision Center is not co-located with Decision Service.
DisableBatchDBOperations	Boolean switch that controls batch database operations.
IntegrationPointGuaranteed RequestTimeout	Guaranteed response time, in milliseconds, for Integration Point requests. (Service Level Guarantee). Zero means don't timeout Integration Point requests - suitable for debugging only.
ModelDSName	The JNDI name of the datasource used by the Learning Service.
ModelSnapshotDSName	The JNDI name of the datasource for the model snapshots.

Attribute	Description
ModelSnapshotMinAbsCorrelation	Controls whether to snapshot all correlation rows or to set a minimum correlation value for snapshots. The default value of 0.000001 prevents snapshots of very small value correlation rows. Set the value to 0 to snapshot all correlation rows.
ModelSnapshotNumberOfBins	Controls the number of bins for model snapshots. Numeric attribute values are automatically binned, or assigned to numeric ranges. The default number of bins is 5. To achieve greater resolution of your numeric data, increase the number of bins. Note that for the same numeric attribute, Oracle RTD creates different bins in different time windows. Therefore, it is unlikely that you will be able to join numeric attribute values across time windows.
SystemDSName	The JNDI name of the datasource.
WorkerThreadPoolSize	The number of threads used for general purpose maintenance activities, not for normal Integration Point request processing. Maintenance activities include model maintenance, session timing, and timed-out request processing.

13.3.3 About OracleRTD > SDClusterPropertyManager > Cluster

The SDClusterPropertyManager > Cluster MBean has the following attributes:

Attribute	Description
BatchManagerInitialWait	The number of milliseconds to wait when the server first starts up before trying to start the Batch Manager Service.
BatchManagerRestartWait	The number of milliseconds to wait after a computer fails or leaves the cluster before trying to restart the Batch Manager Service.
ChoiceHistoryCleanupChunkSize	The chunk size to use when deleting old choice history records.
ChoiceHistoryCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.
GenerateDSCookies	Generate Decision Server HTTP Cookies. Set to `true` to have Decision Server associate Integration Point requests with HTTP sessions, thus causing the Web container to generate container-specific session-affinity cookies.
GenerateDSSessionIdCookie	Generate Decision Service Session-ID Cookies. Set to true to have Decision Service supply a cookie named ORTD_DS_SessionID identifying the DS session serving the current integration point request.
LearningDataStorageCleanupChunkSize	The chunk size to use when deleting old learning data storage records.
LearningDataStorageCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.
LearningServiceInitialWait	The number of milliseconds to wait when the server first starts up before trying to start the Learning Service.

Attribute	Description
LearningServiceRestartWait	The number of milliseconds to wait after a computer fails or leaves the cluster before trying to restart the Learning Service.
OperationalDataCleanupPeriod	The number of hours (fractions are allowed) between cleanup of the operational data (choice history, statistics, learning data storage) in the database.
RequireIntegrationPointAuthorization	True if Integration Points can only be called by security principals granted the \\"decision_service\:normal\\" action on the containing Inline Service.
RestrictDSClients	True if the hosts that can send Decision Service requests is restricted to a fixed list of trusted IP addressees.
StatisticsCleanupChunkSize	The chunk size to use when deleting old statistic records.
StatisticsCleanupThrottle	A number between 0.1 and 1, inclusive. Higher throttle corresponds to higher speed.
TrustedDSClients	List of host IP addresses from which Decision Service requests will be accepted. Port is optional, separated from IP by ':'. Entries are separated by ';'.

Adding Trusted Decision Service Clients

If any host can send Decision Service requests, set RestrictDSClients to False (default value is True).

If you want to restrict the hosts allowed to send Decision Service requests hosts:

- Set RestrictDSClients to True
- For TrustedDSClients, enter a list of the host IP addresses of the client hosts allowed to send Decision Service requests

This change should be propagated to all existing instances, and should be effectively immediately.

13.3.4 About OracleRTD > SDClusterPropertyManager > Deployment

The SDClusterPropertyManager > Deployment MBean has the following attribute:

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

13.3.5 About OracleRTD > SDCluster > SDManagement

Attributes and operations at the SDCluster level are meant to manage cluster-level features.

The SDCluster > SDManagement MBean has the following attributes:

Attribute	Description
BatchManager	Administers the Batch Manager Service attributes.

Attribute	Description
DeploymentStates	Allows the setup and ordering of deployment states.
InlineServiceManager:	Manages deployed Inline Services.
LearningService	Administers the Learning Service attributes.
Members	Members of the cluster. Each Member is listed through this attribute. Member is used to manage local server properties.
Properties	Cluster properties configuration.
Security	Security Manager.

13.4 MBeans for Oracle Real-Time Decisions Member-Level Management

This section provides information about management at the member level.

This section contains the following topics:

- Section 13.4.1, "About OracleRTD > SDManagement > SDPropertyManager"
- Section 13.4.2, "About OracleRTD > SDPropertyManager > Performance Monitoring"
- Section 13.4.3, "About OracleRTD > SDPropertyManager > Misc"
- Section 13.4.4, "About OracleRTD > Server > DecisionService"
- Section 13.4.5, "About OracleRTD > Server > SDManagement"
- Section 13.4.6, "About OracleRTD > Server > BatchAgent"
- Section 13.4.7, "About OracleRTD > Server > BatchManager"
- Section 13.4.8, "About OracleRTD > Server > BatchManager > Proxy > BatchManagerProxy"

13.4.1 About OracleRTD > SDManagement > SDPropertyManager

The SDManagement > SDPropertyManager MBean has the following attributes:

Attribute	Description
PerformanceMonitoring	Performance counter properties.
Misc	Miscellaneous properties.

13.4.2 About OracleRTD > SDPropertyManager > Performance Monitoring

The SDPropertyManager > Performance Monitoring MBean has the following attributes:

Attribute	Description
DSPerfCounterAppend	If true, performance data is appended to an existing file, if any. Otherwise, any existing file is overwritten when the server restarts.
DSPerfCounterEnabled	Enables the writing of DS performance counters. This should not be enabled indefinitely, because the file grows without limit.

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

For more information about using performance monitoring, see Chapter 11, "Performance Monitoring".

13.4.3 About OracleRTD > SDPropertyManager > Misc

The SDPropertyManager > Misc MBean has the following attributes:

Attribute	Description
BatchAgentEnabled	Whether or not Batch Agent should run in this instance.
BatchManagerEnabled	Whether or not Batch Manager should run in this instance.
DecisionCenterEnabled	Whether or not Decision Center should run in this instance.
DecisionServiceEnabled	Whether or not Decision Service should run in this instance.
LearningServiceEnabled	Whether or not Learning Service should run in this instance.
WorkerThreadPoolSize	The number of threads used for general purpose maintenance activities, not for normal Integration Point request processing. Maintenance activities include model maintenance, session timing, and timed-out request processing.

13.4.4 About OracleRTD > Server > DecisionService

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

Attribute	Description
CurrentReceivedRequestsForwarded	Number of requests that were forwarded from other servers to this server, and which have not yet been completely processed by this server.
CurrentRequestsForwarded	Number of requests that have been forwarded from this server to other servers, and for which no acknowledgment has yet been received to indicate that the request has been processed by the forwarded-to server.
CurrentRequestsRunning	Number of Integration Point requests that are currently being processed by Inline Services.
CurrentSessions	Number of Decision Service sessions still open.
PeakReceivedRequestsForwarded	Largest number of received requests forwarded.
PeakRequestsForwarded	Largest number of requests forwarded.
RequestsForwarded	Total number of requests forwarded to another server in the cluster.
TimedOutRequests	Total number of requests that have timed out.
TotalReceivedRequestsForwarded	Total number of received requests forwarded.

Attribute	Description
TotalRequests	Total number of requests seen since the server started.
TotalRequestsForwarded	Total number of requests forwarded.
TotalSessions	Total number of Decision Service sessions created.

13.4.5 About OracleRTD > Server > SDManagement

The Server > SDManagement MBean has the following attributes:

Attribute	Description
Properties	Properties configuration.

13.4.6 About OracleRTD > Server > BatchAgent

The Server > BatchAgent MBean has the following attributes:

Attribute	Description
ActiveBatches	List of all batch jobs currently running on this batch agent, paused, or waiting to run. The list could be empty.
BatchNames	List of batches registered with this batch agent.

13.4.7 About OracleRTD > Server > BatchManager

The Server > BatchManager MBean has the following attributes:

Attribute	Description
ActiveBatches	List of brief status information for all batch jobs currently running, paused, or waiting to run. The list could be empty.
BatchNames	List of batches registered with the batch framework.

13.4.8 About OracleRTD > Server > BatchManager > Proxy > BatchManagerProxy

The Server > BatchManager > Proxy > BatchManagerProxy MBean has the following attribute:

Attribute	Description
BatchManagerLocation	The location where Batch Manager is running.

13.5 MBeans for Managing Inline Services

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

This section contains the following topics:

- Section 13.5.1, "About OracleRTD > SDManagement > InlineServiceManager"
- Section 13.5.2, "About OracleRTD > InlineServiceManager > [Inline Service.Deployment State]"
- Section 13.5.3, "Invoking Maintenance Operations"

13.5.1 About OracleRTD > SDManagement > InlineServiceManager

Each deployed Inline Service is displayed under the InlineServiceManager MBean.

The SDManagement > InlineServiceManager MBean has the following attribute:

Attribute	Description
InlineServices	List of deployed inline services.

The SDManagement > InlineServiceManager MBean has the following operation:

refreshMBeans()

Removes MBeans for applications no longer in the database, and creates MBeans for new ones.

removeAllServices()

Removes all Inline Services (loaded, loadable, failed).

13.5.2 About OracleRTD > InlineServiceManager > [Inline Service.Deployment State]

InlineServiceManager MBeans can be viewed by choosing the name of an Inline Service and a Deployment State, for example: DC_Demo.Development.

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

Attribute	Description
DeploymentState	Development, QA, or Production.
LockStatus	The lock status for the Inline Service.
ServiceId	The service ID for the Inline Service.
Status	Failed, Inactive, or Loadable.

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

unlockService()

Unlocks this service.

removeService()

Stops an Inline Service in this server and removes the service from the database.

flushStatistics()

Flushes all of the statistics for this service to the database.

makeLoadable()

Does a test load on this server and, if successful, marks the service loadable.

deleteStatistics()

Flushes and deletes all of the statistics for this service from the database.

deleteChoiceHistory()

Deletes all of the choice history for this service from the database.

deleteAllOperationalData()

Deletes all of the operational data for this service from the database. This includes choice history, statistics, and the study.

> **Note:** The prediction model data remains in memory for a short time after a user runs the deleteAllOperationalData() operation.

deleteStudy()

Removes the study for this service.

13.5.3 Invoking Maintenance Operations

Maintenance Operations appear in a node under an Inline Service when both of the following conditions hold:

- The Inline Service includes one or more Maintenance Operations
- The Inline Service has the Status flag set to **Loadable**

Each Maintenance Operation appears in both of the **BroadcastAsyncOperations** and **DirectBlockingOperations** nodes, under Maintenance Operations.

Operations listed in the **DirectBlockingOperations** node are invoked on the local server only, and they return only after the operation has completed. The returned value will be displayed in a popup dialog. If the operation has return type "void," then "null" will appear. If the operation fails for any reason, a short error message will be displayed in a popup dialog, and a more detailed report can be found in the log of that server.

Operations listed in the **BroadcastAsyncOperations** node are invoked across every node of a cluster. The operation returns immediately with the number of cluster members who received the broadcast. If the cluster has just one node, the operation returns 1, and the invocation is still asynchronous.

The following run-time considerations apply for Maintenance Operations:

- Oracle RTD does not guarantee that all cluster members are notified of Maintenance Operation invocations, although usually they will be.

  For example, if one member of a cluster is down when a Maintenance Operation is invoked, there is no notification to indicate that the cluster member should run the Maintenance Operation when it comes back up.
- Ordering of Maintenance Operations is not guaranteed. For example, if two Maintenance Operations A and B are invoked in sequence, an Inline Service may run B before it runs A, or it may even run them simultaneously.

13.6 MBeans for Deployment States

By viewing the OracleRTD > DeploymentStates MBeans, you can see a list of deployment states that are available on the cluster.

This section contains the following topics:

- Section 13.6.1, "About OracleRTD > SDManagement > DeploymentStates"

- Section 13.6.2, "About OracleRTD > Deployment States > [State]"

13.6.1 About OracleRTD > SDManagement > DeploymentStates

The SDManagement > DeploymentStates MBean has the following attribute:

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

13.6.2 About OracleRTD > Deployment States > [State]

MBeans for a particular Deployment State can be viewed by choosing OracleRTD > Deployment States, then choosing a Deployment State (for example, Development, QA, or Production).

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

Attribute	Description
AllowHotSwapping	Allow hot swapping of Inline Services with this deployment state in Decision Service.
Id	ID of the deployment state.
Name	Name of the deployment state.

> **Note:** If hot swapping is enabled for a deployment state, and an Inline Service is redeployed in the state, the existing Inline Service will remain active until all existing sessions close or timeout. New sessions will be created on the newly deployed Inline Service.

13.7 MBeans for Managing Learning Services

Managing Learning Services on the cluster allows you to check the status of the learning models and perform maintenance on them.

This section contains the following topics:

- Section 13.7.1, "About OracleRTD > Server > LearningService"
- Section 13.7.2, "About OracleRTD > Server > LearningService > Proxy > LearningServiceProxy"
- Section 13.7.3, "About OracleRTD > Learning Server > [Study]"
- Section 13.7.4, "About OracleRTD > Study > [Model.Study]"

13.7.1 About OracleRTD > Server > LearningService

The Server > LearningService MBean has the following attribute:

Attribute	Description
Studies	A list of all Studies running on the Learning Server. The models of a Study are viewed by clicking on a Study.

13.7.2 About OracleRTD > Server > LearningService > Proxy > LearningServiceProxy

The Server > LearningService > Proxy > LearningServiceProxy MBean has the following attribute:

Attribute	Description
LearningServiceLocation	The location where Learning Service is running.

13.7.3 About OracleRTD > Learning Server > [Study]

The Learning Server > [Study] MBeans have the following attributes:

Attribute	Description
Models	Models belonging to this study.
Name	The name of the study.

The Learning Server > [Study] MBeans have the following operation:

CompleteSnapshot()

Saves a snapshot of this study to the database (complete).

Delete()

Deletes study data.

DeleteSnapshot()

Removes this study's snapshot from the database.

IncrementalSnapshot()

Saves a snapshot of this study to the database (delta).

13.7.4 About OracleRTD > Study > [Model.Study]

The Study > [Model.Study] MBeans have the following attributes:

Attribute	Description
Attributes	Names of the model attributes. The names listed here match the attribute names in the session for your Inline Service.
Name	The name of the model.
TimeWindows	List of the ranges of time that have been learned about by this particular model.

The Study > [Study Name] > [Model] MBeans have the following operations:

Delete()

Deletes model data.

DeleteAttributeValue()

Erases model data collected for a value of an attribute. This operation accepts the following parameters:

AttributeName: The name of an attribute.

Value: The value to be deleted.

DeleteAttributeValueRange()

Erases model data collected for a range of values of an attribute. This operation accepts the following parameters:

AttributeName: The name of an attribute.

HighestValue: The highest value to be deleted.

LowestValue: The lowest value to be deleted.

StartNewTimeWindow()

Closes the current time window and starts a new one. Do not use this operation in a production environment, because it may impair future model learning.

13.8 Post-Deployment Management of Inline Services

The deployment of an Inline Service writes two types of data to the SDDB database, **metadata** and **content**.

Inline Service **metadata** describes the underlying elements and structure of the Inline Service, in effect, the framework of the Inline Service.

Inline Service **content** is data that changes at runtime, and consists of the following general types of data:

- (A) Study (a collection of one or more models)

  > **Note:** Each Inline Service's learnings are associated with a study name. If you want to redeploy an Inline Service and restart its learnings, deploy it with a new study name. Different study names can be used for Development, QA, and Production.

- (B) Statistics
- (C) Choice history
- (D) Learning data

The following table shows the JMX MBeans and operations to use when you want to remove an Inline Service or to delete some or all of the content data of an Inline Service.

The Operation column shows the operation to select for the corresponding MBean-related Topic, which is described separately in this chapter. The Actions which relate to content data include one or more letter references to the content types in the preceding bulleted list.

Action	Operation	Topic
Removing an Inline Service *This deletes all the metadata of an Inline Service, but none of the content.*	removeService()	About OracleRTD > InlineServiceManager > [Inline Service.Deployment State]

Action	Operation	Topic
Deleting Inline Service statistics (B)	deleteStatistics()	About OracleRTD > InlineServiceManager > [Inline Service.Deployment State]
Deleting Inline Service choice history (C)	deleteChoiceHistory()	About OracleRTD > InlineServiceManager > [Inline Service.Deployment State]
Deleting all Inline Service operational data (A)+(B)+(C)+(D) ***This deletes all the content data of an Inline Service, but no metadata***.	deleteAllOperationalData()	About OracleRTD > InlineServiceManager > [Inline Service.Deployment State]
Deleting Inline Service study (A)	deleteStudy()	About OracleRTD > InlineServiceManager > [Inline Service.Deployment State]
(Alternate) Deleting Inline Service study (A)	Delete()	About OracleRTD > Learning Server > [Study]
Deleting models from an Inline Service study This deletes a single model from a study (A).	Delete()	About OracleRTD > Study > [Model.Study]

Caution: It is strongly advised that, prior to performing any remove or delete action, you backup the SDDB database.

13.9 System Properties

You can view and set the following system properties:

System Property	Description
ModelDSName	The JNDI name of the datasource used by the Learning Service. Default=SDDS.
ModelSnapshotDSName	The JNDI name of the datasource used by the Learning Service to perform snapshots of its learning models. Default=SDDS.
RestrictClusterMembers	True if the hosts that can be in the cluster is restricted to a fixed list of trusted IP addresses.
RestrictDSClients	True if the hosts that can send Decision Service requests is restricted to a fixed list of trusted IP addresses.
SDGroupName	This is the name of the Oracle RTD cluster, as recognized by Oracle RTD's cluster management.
SystemDSName	This is the name of the Oracle RTD system datasource. Default=SDDS.
TrustedClusterMembers	List of host IP addresses that can join the cluster. Port is optional, separated from IP by ':'. Entries are separated by ';'.
TrustedDSClients	List of host IP addresses from which Decision Service requests will be accepted. Port is optional, separated from IP by ':'. Entries are separated by ';'.

A

System Log and Configuration Files

This appendix provides the names and locations of the Oracle RTD log and configuration files. To ensure the integrity of the information in these files, you should use operating system or other tools to set the appropriate file permissions on the parent directories that contain these files, so that an intruder cannot access them.

This section contains the following topics:

- Section A.1, "Searching and Viewing Server-Side Log Files"
- Section A.2, "Configuring Oracle RTD Server-Side Log Files"
- Section A.3, "Log Files"
- Section A.4, "Configuration Files"

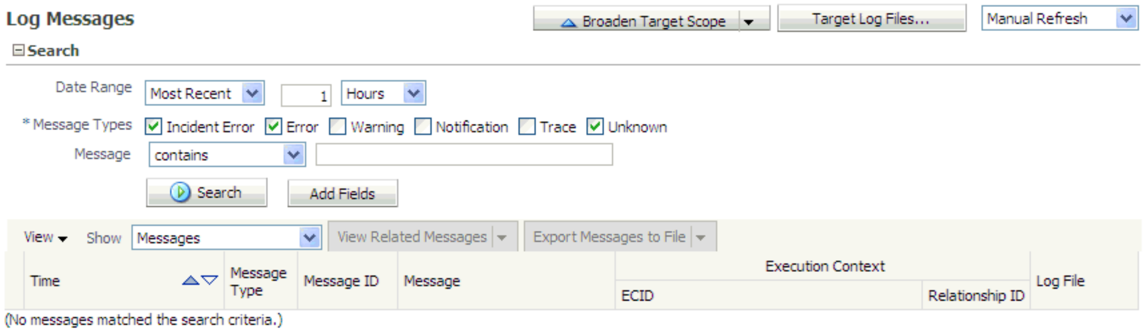
A.1 Searching and Viewing Server-Side Log Files

In the Enterprise Manager of Fusion Middleware Control, you can search and view log files both within and across Oracle Fusion Middleware components. You can also download log files to your local client and view the log files using other tools.

You must first log into Fusion Middleware Control. For details, see Section 2.1.1, "Logging into Fusion Middleware Control."

To access the log files from the Oracle RTD home page, right-click the deployed Oracle RTD application in the navigation pane, and select Logs, then View Log Messages.

The Log Messages screen appears.



Log Messages

Search

Date Range: Most Recent 1 Hours

* Message Types: ☑ Incident Error ☑ Error □ Warning □ Notification □ Trace ☑ Unknown

Message: contains

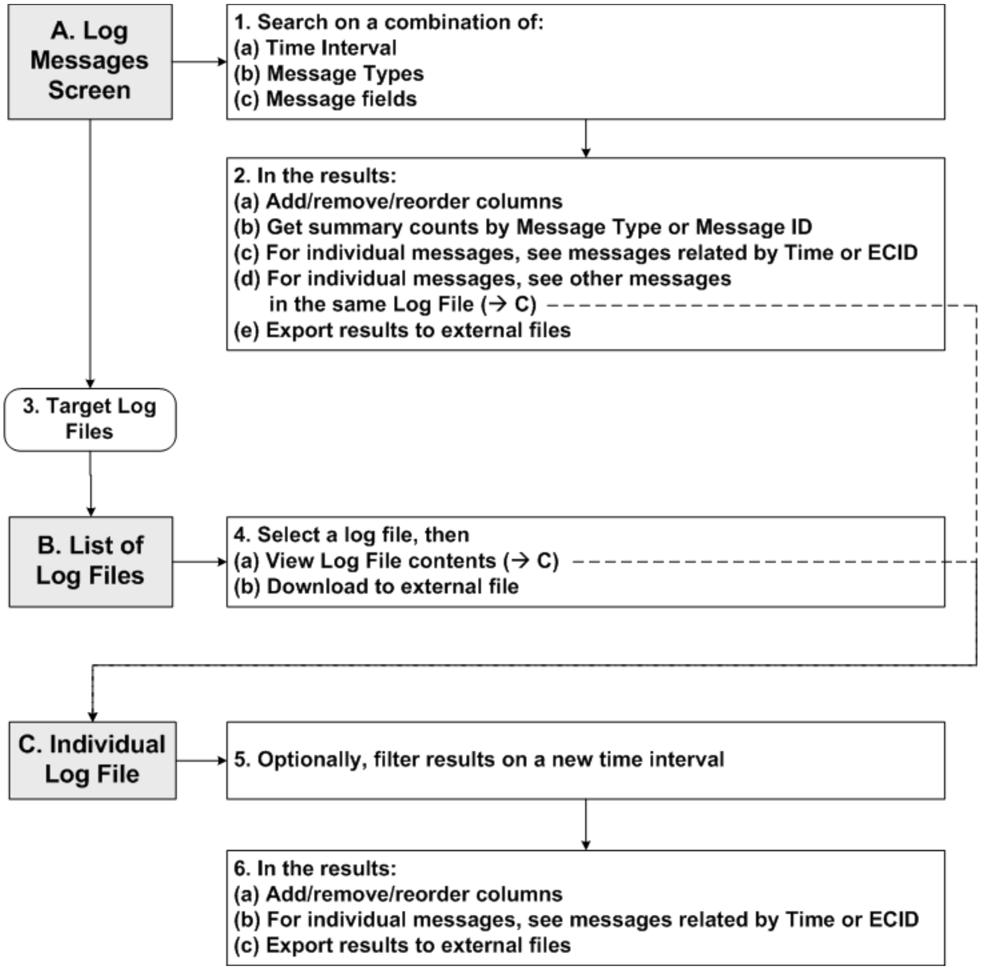
Show: Messages

Time	Message Type	Message ID	Message	Execution Context		Log File
				ECID	Relationship ID	

(No messages matched the search criteria.)

Figure A–1 shows an overview of the main Oracle Fusion Middleware operations to search and view log files.

Figure A–1 Main Oracle Fusion Middleware Log File Search and View Operations


A. Log Messages Screen
1. Search on a combination of:
(a) Time Interval
(b) Message Types
(c) Message fields
2. In the results:
(a) Add/remove/reorder columns
(b) Get summary counts by Message Type or Message ID
(c) For individual messages, see messages related by Time or ECID
(d) For individual messages, see other messages in the same Log File (→ C)
(e) Export results to external files
3. Target Log Files
B. List of Log Files
4. Select a log file, then
(a) View Log File contents (→ C)
(b) Download to external file
C. Individual Log File
5. Optionally, filter results on a new time interval
6. In the results:
(a) Add/remove/reorder columns
(b) For individual messages, see messages related by Time or ECID
(c) Export results to external files

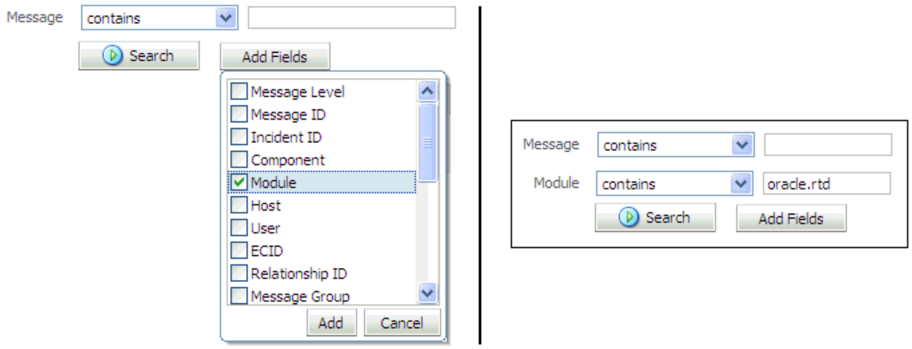
For general information about searching and viewing log files, see the section "Viewing and Searching Log Files" in *Oracle Fusion Middleware Administrator's Guide.*

The following notes are more specific to Oracle RTD users:

1. The list of log files includes the Managed Server log file, which has the general name `*server_name*.log`.

   The Oracle RTD runtime log files have the general name `*server_name*-diagnostic*[-<n>]*.log`.

2. To search for files that are specific to Oracle RTD, first add the field Module to the search fields, enter **oracle.rtd** in the Module search box, then click Search.



Message: contains

Module: contains oracle.rtd

A.2 Configuring Oracle RTD Server-Side Log Files

You can configure the following properties of Oracle RTD server-side log files:

- The name and location of log files.
- The size of log files.

  You can specify that a new file is created either when the log file reaches a certain size or when a particular time is reached.
- The level of information written to log files.

  This enables you to control the type and level of detail of information written to log files, by specifying message type and message level. The most common message types are Error, Warning, Notification, and Trace. You can set the message level of Notification and Trace files to control the granularity of message detail.
- The format of the log files.
- The Locale encoding.

For more information about configuring log files, see the section "Configuring Settings for Log Files" in *Oracle Fusion Middleware Administrator's Guide.*

A.3 Log Files

Oracle RTD provides both server and client logs. In addition, there are log files specific to Eclipse and application servers.

This section contains the following topics:

- Section A.3.1, "Main Oracle RTD Log Files"
- Section A.3.2, "Log Files for Oracle RTD Client Tools"
- Section A.3.3, "Server-Side Log Files"
- Section A.3.4, "Eclipse Log File"

A.3.1 Main Oracle RTD Log Files

Table A–1 shows the two main Oracle RTD log files. The Oracle RTD Server log is the main log to use for troubleshooting problems.

Table A–1 Main Oracle RTD Logs

Log Type	Default Location
Oracle RTD Server log	`RTD_RUNTIME_HOME/log/server_name-diagnostic[-<n>].log`
Oracle RTD Client log	`RTD_HOME/log/client.log`

- `RTD_HOME` is the directory into which you extract the client-side Oracle RTD files.
- `RTD_RUNTIME_HOME` is the directory into which you install runtime Oracle RTD, typically `<mw_home>/user_projects/domains/domain_name/servers/server_name/`.

A.3.2 Log Files for Oracle RTD Client Tools

In addition to the Oracle RTD Client log, Oracle RTD maintains the following client tool log files:

- `RTD_HOME/scripts/SDDBTool.log`
- `RTD_HOME/log/loadgen.csv`

For Decision Studio log messages, see Section A.3.4, "Eclipse Log File."

A.3.3 Server-Side Log Files

This section lists the server-side log files under WebLogic.

```
RTD_RUNTIME_HOME/logs/server_name.log
RTD_RUNTIME_HOME/logs/server_name-diagnostic[-<n>].log
RTD_RUNTIME_HOME/logs/server_name.out
```

A.3.4 Eclipse Log File

The Eclipse log file, used for logging Decision Studio log messages, is `Decision_Studio_Workspace/.metadata/.log`.

Setting Logging Levels for Eclipse

To set the logging levels for Eclipse, edit the following file:

```
RTD_HOME\eclipse\plugins\com.sigmadynamics.studio_
11.1.1\etc\eclipse-log.properties
```

To adjust logging levels, set the values to `true` or `false`. The default settings are as follows:

- debug=false
- info=true
- warn=true
- error=true
- fatal=true
- trace=false

A.4 Configuration Files

This section lists the Oracle RTD configuration files in Release 11g.

Oracle RTD Configuration Files for WebLogic

```
<Oracle_BI_Directory>/bifoundation/jee/RTD.ear/APP_INF/lib/etc/sdconfig.xml
```

B

Upgrading and Patching Oracle Real-Time Decisions

This appendix introduces the concepts of upgrading and patching Oracle RTD, and specifies where details of upgrading and patching Oracle RTD may be found.

Note: Before you start any patch or upgrade process, you should refer to the release notes that apply either to your patch or to the release levels of your upgrade.

For information about upgrading your existing Oracle Real-Time Decisions 3.0.0.1 environment to Oracle Real-Time Decisions 11*g*, see "Upgrading Oracle Real-Time Decisions" in *Oracle Fusion Middleware Upgrade Guide for Oracle Business Intelligence*.

Patching is the process of copying a small collection of files over an existing instance (also referred to as an "in-place installation"). A patch is normally associated with a particular version of an Oracle product and involves updating from one minor version of the product to a newer minor version of the same product. A patch set is a single patch that contains a collection of patches designed to be applied together.

If you plan to install any Oracle Real-Time Decisions patch set, see *Oracle Fusion Middleware Patching Guide*.