

**Oracle® Real-Time Decisions for Siebel  
E-Commerce**

Installation and Reference Guide

Version 2.2.1

**E12261-01**

May 2008

Copyright © 2008, Oracle. All rights reserved.

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

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

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

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

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

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

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

---

---

# Contents

<b>Preface</b> .....	vii
Audience .....	vii
Documentation Accessibility .....	vii
Related Documents .....	viii
Conventions .....	viii
 <b>1 Installation of Oracle Real-Time Decisions for Siebel E-Commerce</b>	
1.1 Configuring JDBC Data Source for Access to Siebel OLTP Data .....	1-1
1.1.1 Creating WebSphere 6.1 Data Sources .....	1-2
1.1.2 Creating BEA WebLogic 9.2 Data Sources .....	1-2
1.1.3 Creating Oracle Application Server 10.1.3 Data Sources .....	1-3
1.2 Importing the RTD Siebel ECommerce Inline Service into Oracle RTD Decision Studio .....	1-3
1.3 Verifying the Setup of the Siebel Data Source .....	1-4
1.4 Deploying the RTD Siebel ECommerce Inline Service to the Oracle RTD Server .....	1-4
1.5 Specifying the List of Trusted Hosts for Decision Service Requests .....	1-4
1.6 Testing the Deployed Inline Service Via Web Service Calls .....	1-5
1.7 Setting Up the RTD_WEB_HISTORY Table (Optional) .....	1-6
 <b>2 Overview of Oracle Real-Time Decisions (Oracle RTD)</b>	
2.1 Introduction to the Oracle RTD Decision Process .....	2-1
2.2 Integration Points Between External Applications and Oracle RTD .....	2-4
2.3 Overview of Oracle RTD Integration with External Applications .....	2-4
2.4 The Oracle RTD Decision Process .....	2-6
2.5 More About the Oracle RTD Decision Process Elements .....	2-7
2.5.1 Performance Goals .....	2-7
2.5.2 Decisions .....	2-8
2.5.3 Choice Groups and Choices .....	2-8
2.5.4 Eligibility Rules .....	2-9
2.5.5 Filtering Rules .....	2-9
2.5.6 Scoring Rules .....	2-9
2.5.7 Models .....	2-9
2.5.7.1 Model Usage Summary .....	2-10
2.5.7.2 Choice Models .....	2-10
2.5.7.3 Choice Event Models .....	2-10
2.6 General Oracle RTD Elements and Features .....	2-11

2.6.1	Data Sources .....	2-11
2.6.2	Entities.....	2-11
2.6.3	Functions.....	2-12
2.7	Oracle RTD Inline Services .....	2-12
2.8	Introduction to Oracle RTD Decision Studio.....	2-13
2.9	Analytic Reports and the Oracle RTD Decision Center .....	2-14

### 3 RTD Siebel ECommerce Inline Service Elements

3.1	Overview of Oracle RTD Integration with Siebel E-Commerce .....	3-1
3.2	Data Sources .....	3-3
3.2.1	Accounts DS .....	3-4
3.2.2	Address DS .....	3-4
3.2.3	Assets DS (Multiple Rows).....	3-5
3.2.4	Catalog Category DS.....	3-5
3.2.5	Catalog DS .....	3-5
3.2.6	Contact Address DS .....	3-5
3.2.7	Contact DS .....	3-6
3.2.8	Image DS .....	3-6
3.2.9	Industry DS.....	3-6
3.2.10	List of Values DS.....	3-7
3.2.11	Product Line DS .....	3-7
3.2.12	Products DS .....	3-7
3.2.13	Promo Category Catalog DS .....	3-7
3.2.14	Promotions DS.....	3-8
3.2.15	Web History DS .....	3-8
3.3	Entities .....	3-8
3.3.1	Session Entity .....	3-10
3.3.1.1	Writing Output to RTD_WEB_HISTORY Table During Session Cleanup .....	3-10
3.3.2	Account Entity.....	3-11
3.3.3	Assets Entity .....	3-11
3.3.4	Catalog Entity.....	3-12
3.3.5	Category Entity .....	3-13
3.3.6	Contact Entity.....	3-13
3.3.7	Industry Entity .....	3-14
3.3.8	Products Entity.....	3-14
3.3.9	Promo Category Catalog Entity.....	3-15
3.3.10	Promotions Entity.....	3-15
3.3.11	Promotions List Entity .....	3-16
3.3.12	Raw Assets Entity .....	3-17
3.3.13	Raw Web Interactions Entity .....	3-18
3.3.14	Web Interaction Entity .....	3-19
3.4	Informants.....	3-21
3.4.1	Initial Context.....	3-21
3.4.2	Offer Clicked .....	3-22
3.4.3	Presented Offers.....	3-22
3.4.4	Update Customer Profile.....	3-23
3.4.5	Close Session .....	3-23

3.5	Advisors .....	3-24
3.5.1	Get Promotions .....	3-24
3.6	Choice Groups and Choices .....	3-24
3.6.1	Promotions Choice Group .....	3-24
3.6.2	Siebel Promotions Choice Group (Dynamic Choices).....	3-25
3.6.3	RTD Promotions Choice Group (Static Choices) .....	3-27
3.7	Decisions .....	3-28
3.8	Models .....	3-28
3.9	Functions .....	3-28

## **4 Configuring the RTD Siebel ECommerce Inline Service**

4.1	Utilizing the Category/Catalog Eligibility Rules.....	4-1
4.2	Modifying Promotion Attributes Sent to Client.....	4-2
4.3	Adding More Oracle RTD Data Sources .....	4-2
4.4	Adding Attributes to Existing Entities.....	4-3
4.5	Modifying Scoring Logic .....	4-4
4.6	Enabling the RTD_WEB_HISTORY Table .....	4-4



---

# Preface

This document describes the features of the RTD Siebel ECommerce Inline Service, that integrates with the Siebel E-Commerce application.

## Audience

This document is intended for the following Oracle RTD users:

- Technical users configuring Inline Services using Decision Studio
- Business users of Decision Center
- Administrators

## Documentation Accessibility

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

### Accessibility of Code Examples in Documentation

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

### Accessibility of Links to External Web Sites in Documentation

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

### TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

## Related Documents

For more information, see the following documents in the Oracle Real-Time Decisions platform Version 2.2.1 documentation set:

- *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*
- *Oracle Real-Time Decisions Decision Studio Reference Guide*
- *Oracle Real-Time Decisions Decision Center User Guide*
- *Oracle Real-Time Decisions Release Notes*
- *Oracle Real-Time Decisions New Features Guide Version 2.2.1*

## Conventions

The following text conventions are used in this document:

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



---

# Installation of Oracle Real-Time Decisions for Siebel E-Commerce

This chapter describes the steps required to install Oracle Real-Time Decisions for Siebel E-Commerce.

As prerequisites, you must have installed one of the Oracle RTD supported J2EE application servers, and you must have successfully deployed the Oracle RTD server on to this application server. Before starting the installation, verify that Oracle RTD is running and that the server logs do not contain any errors.

For information on installing the Oracle RTD Server, please refer to *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*, which is available with the Oracle RTD platform software.

The chapter describes how to set up the Oracle RTD-specific portion of the overall Siebel E-Commerce application, that is, the RTD Siebel ECommerce Inline Service.

---

**Note:** Details on the integration to Oracle RTD from the front end application can be found in the *Siebel E-Commerce Administration Guide*, in the section "Setting Up Hot Offers for Siebel E-Commerce."

---

Setting up the Inline Service for the Siebel E-Commerce application involves the following tasks:

- [Section 1.1, "Configuring JDBC Data Source for Access to Siebel OLTP Data"](#)
- [Section 1.2, "Importing the RTD Siebel ECommerce Inline Service into Oracle RTD Decision Studio"](#)
- [Section 1.3, "Verifying the Setup of the Siebel Data Source"](#)
- [Section 1.4, "Deploying the RTD Siebel ECommerce Inline Service to the Oracle RTD Server"](#)
- [Section 1.5, "Specifying the List of Trusted Hosts for Decision Service Requests"](#)
- [Section 1.6, "Testing the Deployed Inline Service Via Web Service Calls"](#)
- [Section 1.7, "Setting Up the RTD\\_WEB\\_HISTORY Table \(Optional\)"](#)

## 1.1 Configuring JDBC Data Source for Access to Siebel OLTP Data

The following JDBC data sources are required for this application:

Data Source Name	Type	Description
SDDS	System	Stores the deployed projects, run-time sessions and model learnings. This data source should have already been configured as part of the Oracle RTD server setup.
Siebel	Source	Used to fill the Oracle RTD session entities such as the customer and asset entities from the Siebel OLTP.

Create the Siebel data source by selecting the steps appropriate to the application server you are using, from the following list:

- [Section 1.1.1, "Creating WebSphere 6.1 Data Sources"](#)
- [Section 1.1.2, "Creating BEA WebLogic 9.2 Data Sources"](#)
- [Section 1.1.3, "Creating Oracle Application Server 10.1.3 Data Sources"](#)

### 1.1.1 Creating WebSphere 6.1 Data Sources

For instructions on how to install and deploy Oracle RTD Server through the WebSphere admin console, see the Oracle RTD platform document *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*, Section 4.

The following steps describe how to create the Siebel data source and register it with the Oracle RTD server:

1. Create the Siebel data source within WebSphere, as described in Section 7.2 of *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*. The data source parameters required (database host, name, port, username, password, and so on) are those of the Siebel OLTP database.

In section 7.2.1, steps 13b and 13c, be sure to enter **Siebel** as the data source name and the JNDI name.

This data source will be used by Oracle RTD to query the Siebel OLTP and to populate information entities such as the customer, agent and products owned.

2. Create resource references to the Siebel data source. This process is described in Section 7.2 of *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*.
3. Restart the application server.

### 1.1.2 Creating BEA WebLogic 9.2 Data Sources

For instructions on how to install and deploy Oracle RTD server through the WebLogic admin console, see the Oracle RTD platform document *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*, Section 5.

The following steps describe how to create the Siebel data source and register it with the Oracle RTD server:

1. Create the Siebel data source within WebLogic, as described in Section 7.3 of *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*. The data source parameters required (database host, name, port, username, password, and so on) are those of the Siebel OLTP database.

In section 7.3.2, steps 4a and 4b, be sure to enter **Siebel** as the data source name and the JNDI name.

This data source will be used by Oracle RTD to query the Siebel OLTP and to populate information entities such as the customer, agent and products owned.

2. Create resource references to the Siebel data source. This process is described in Section 7.3 of *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*.
3. Restart the application server.

### 1.1.3 Creating Oracle Application Server 10.1.3 Data Sources

For instructions on how to install and deploy Oracle RTD server through the Oracle Enterprise Manager console, see the Oracle RTD platform document *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*, Section 3.

The following steps describe how to create the Siebel data source and register it with the Oracle RTD server:

1. Create the Siebel data source within OC4J, as described in Section 7.1 of *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*. The data source parameters required (database host, name, port, username, password, and so on) are those of the Siebel OLTP database.

In section 7.1.2, steps 11a and 11b, be sure to enter **Siebel** as the data source name and the JNDI name.

This data source will be used by Oracle RTD to query the Siebel OLTP and to populate information entities such as the customer, agent and products owned.

2. Create resource references to the Siebel data source. This process is described in Section 7.1 of *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*.
3. Restart the application server.

## 1.2 Importing the RTD Siebel ECommerce Inline Service into Oracle RTD Decision Studio

The RTD Siebel ECommerce Inline Service is packaged in a zip file which contains XML and Java files organized in a directory structure. To load, view, and edit the Inline Service, use the Oracle RTD client tool Decision Studio.

To set up the RTD Siebel ECommerce Inline Service, you must import the Inline Service project into Decision Studio, as follows:

1. Create a directory for your Inline Services, for example, `C:\RTD_ILS`.

This directory will be referred to as *RTD\_ILS\_HOME* in this documentation.

2. From the Oracle RTD Applications software directory, *Oracle Real-Time Decisions for Siebel E-Commerce*, unzip the application zip file, *RTD\_Siebel\_ECommerce.zip*, into *RTD\_ILS\_HOME*.

This creates the directory *RTD\_Siebel\_ECommerce* under *RTD\_ILS\_HOME*.

3. In Decision Studio, select **File > Import**.
4. From the Import window, select **Existing Projects into Workspace**.
5. If not already visible in the Projects area of the Import Projects dialog box, click the **Browse** button, and locate *RTD\_Siebel\_ECommerce* under *RTD\_ILS\_HOME*, for example `C:\RTD_ILS\RTD_Siebel_ECommerce`.

6. Select the directory `RTD_Siebel_ECommerce` in the file dialog box and click Ok.
7. Confirm that the **RTD\_eCommerce** Inline Service project name appears in the project window and is checked True.
8. Click the Finish button to load the project into Decision Studio.

## 1.3 Verifying the Setup of the Siebel Data Source

The next step is to verify that the Siebel data source is accessible by the Oracle RTD server. This verifies that you have set up the Oracle RTD data source properly as described in [Section 1.1, "Configuring JDBC Data Source for Access to Siebel OLTP Data."](#)

Perform the following steps:

1. In the `RTD_eCommerce` Inline Service project, expand the Data Sources folder and double click **Assets DS**.
2. Click the Import button in the configuration window.
3. Select the active Oracle RTD server from the dialog box that pops up and click Next.

If the JDBC data source was configured correctly, the application enables you to select **Siebel** as a data source and then to display the list of tables in the OLTP.

If this is not the case, check the configuration of your JDBC data source, and if necessary perform the configuration again.

4. Click the Cancel button, since there is no need to re-import the data source.

## 1.4 Deploying the RTD Siebel ECommerce Inline Service to the Oracle RTD Server

The RTD Siebel ECommerce Inline Service is now ready to be deployed to the Oracle RTD server, as follows:

1. Click Deploy, located in the Decision Studio toolbar.
2. If multiple projects are open in Decision Studio, ensure that the correct Inline Service project, **RTD\_eCommerce**, is selected.
3. Ensure that the deployment state is set to Development.
4. Click Deploy.

If the Inline Service deploys successfully, the message "RTD\_eCommerce deployed successfully" appears in the bottom status bar.

Before the Oracle RTD server can accept requests, the list of trusted hosts must be updated to include IP addresses where requests will originate. The instructions to achieve this goal appear in the section that follows.

## 1.5 Specifying the List of Trusted Hosts for Decision Service Requests

If both the E-Commerce server and the Oracle RTD server are installed on the same physical computer, this section can be bypassed. If the servers reside on different computers (or belong under different IP addresses), the IP address of the E-Commerce server must be added to a list of trusted hosts for Oracle RTD Decision Service requests.

Similarly, if Oracle RTD Decision Studio (or Oracle RTD Load Generator) is located on a different computer from that of the Oracle RTD Server, and you want to send sample requests using the Test View in Decision Studio, then the IP address of the computer that hosts either or both of Decision Studio and Load Generator will also need to be added to the list of trusted hosts. The list is stored within the Oracle RTD server and maintained through Oracle RTD's JMX administration console (JConsole) as described later in this section.

To specify a list of trusted hosts for Oracle RTD Decision Service requests, perform the following steps:

1. If you are using OC4J or WebLogic, open JConsole by running `JAVA_HOME/bin/jconsole.exe`.

If you are using WebSphere, run the batch script you created during JConsole configuration.

For more information about accessing JConsole, see the Oracle RTD platform document *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*.

2. Click the **Remote** tab.
3. Enter the appropriate port number (typically 12345) and the administrator credentials that you created during installation.
4. Click Connect.
5. Click the **MBean** tab.
6. Navigate the path `OracleRTD > SDClusterPropertyManager > Cluster MBean`.
7. Ensure that the **RestrictDSClients** attribute is set to true.  
This attribute ensures that the Decision Service only accepts requests from its own host, or from the list of hosts identified in the **TrustedDSClients** attribute.
8. Update the **TrustedDSClients** attribute to include a semicolon-separated list of IP addresses of the hosts from which you want Decision Service to accept requests.

You must specify IP addresses. Do not specify host names.

## 1.6 Testing the Deployed Inline Service Via Web Service Calls

When the data source connectivity has been verified, and the Inline Service has been deployed, the next step is to do a dry run of the application using the Oracle RTD Decision Studio's built-in Test View.

After integration has been set up with the E-Commerce front end, integration calls to Oracle RTD can also be initiated there as well.

1. Within the JMX console, set the logging level to DEBUG.

Debug statements have been incorporated into the inline service that help trace the execution of each informant. Setting the log mode to DEBUG will allow the user to see these statements in the Oracle RTD server log.

2. Navigate to the Test tab on the lower quadrant of Oracle RTD Decision Studio.

If this tab is hidden, from the Oracle RTD Decision Studio's main menu, select `Window > Show View > Test` to expose it.

3. Select **Update Customer Profile** from the Integration Point dropdown.
4. For this test, fill in any value for the **Web Session Id** attribute.

5. From the **Account Id** attribute, select a real ROW\_ID value for any account record found in the Siebel OLTP.

Leave the other attributes blank.

6. Click the Play button and a web services call is made to the Oracle RTD Server.

The Log tab displays the output results. Verify that customer values returned in the output are the same values found in the OLTP. For example, validate that the customer's State and Zipcode is correct.

7. Verify that customer values returned in the output are the same values found in the OLTP.

For example, validate that the customer's State and Zipcode are correct.

8. If the test fails, examine the exception stack trace from either the server command window, if one exists, or from the Oracle RTD server.log file.

This log contains information useful in debugging the problem.

For typical locations of the Oracle RTD server.log file, see section 6.1.2 of *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*.

## 1.7 Setting Up the RTD\_WEB\_HISTORY Table (Optional)

The RTD\_WEB\_HISTORY table is a unique table created in the RTD schema (SDDS) specifically for the Siebel E-Commerce solution. When utilized, it stores a transaction of each web session started in Oracle RTD along with key data about the session itself. Each time the session closes or times out, Oracle RTD writes a record to this table.

The logic to write to this table can be found in the "cleanup logic" section of the Session entity.

To create and enable the RTD\_WEB\_HISTORY table, perform the following steps:

1. From the *RTD\_ILS\_HOME\RTD\_Siebel\_ECommerce* directory created for the RTD Siebel E-Commerce Inline Service, navigate to the **etc** folder.

Inside the **etc** folder, there are several SQL scripts entitled **Create RTD Web History...**

2. Select the appropriate script for the database type used for your Oracle RTD database.
3. Run the script to create the RTD\_WEB\_HISTORY table in Oracle RTD's SDDS database.

To run the script, use a SQL editor appropriate to your database type.

4. After the table is created, validate that the RTD server can see this table following a similar procedure to the one outlined in [Section 1.3, "Verifying the Setup of the Siebel Data Source."](#)
5. To enable Oracle RTD to write to this table through the supplied logic, the Application parameter **Enable RTD Web History** must be set to True in the Inline Service.

Note that in the released version of the RTD Siebel ECommerce Inline Service, the value of **Enable RTD Web History** is set to True.

After the RTD\_WEB\_HISTORY table is created, a record for each Oracle RTD session will be stored in this table, which can then be used by the Web Interaction Entity.

If you need more columns in this table, you can modify the SQL script to add the additional columns. You will also need to modify the Log Writer and Insert Message Result functions included as part of the Inline Service, so that Oracle RTD can write data to these new columns.





---

# Overview of Oracle Real-Time Decisions (Oracle RTD)

Oracle Real-Time Decisions (Oracle RTD) enables you to develop adaptive enterprise software solutions. These solutions are adaptive because they use rules and predictive models to continuously learn from business process transactions as those transactions are executing. By continuously learning in real time, the adaptive solutions that you develop can optimize the outcome of each transaction and of the associated business process.

This chapter presents an overview of Oracle RTD, and of the Oracle RTD features that are used when Oracle RTD is integrated with external applications.

For more detailed information about Oracle RTD, see [Related Documents](#) in the [Preface](#) chapter.

This chapter consists of the following topics:

- [Section 2.1, "Introduction to the Oracle RTD Decision Process"](#)
- [Section 2.2, "Integration Points Between External Applications and Oracle RTD"](#)
- [Section 2.3, "Overview of Oracle RTD Integration with External Applications"](#)
- [Section 2.4, "The Oracle RTD Decision Process"](#)
- [Section 2.5, "More About the Oracle RTD Decision Process Elements"](#)
- [Section 2.6, "General Oracle RTD Elements and Features"](#)
- [Section 2.7, "Oracle RTD Inline Services"](#)
- [Section 2.8, "Introduction to Oracle RTD Decision Studio"](#)
- [Section 2.9, "Analytic Reports and the Oracle RTD Decision Center"](#)

## 2.1 Introduction to the Oracle RTD Decision Process

The heart of Oracle RTD is a "decision engine" that helps users make decisions, by recommending the best options when they make their choices.

To illustrate the principles of the decision process and how these are incorporated in the Oracle RTD "decision engine", consider a common real-world decision: whether or not to accept a job offer from one of several companies that you have been investigating?

The data involved in the decision making process can be extensive. For example, a small subsection of job-related data to collect and evaluate could be:

- Company offering the job

- Job title
- Location
- Salary
- Promotion prospects

Company	Job	Location	Salary	Promotion Prospects
VeriLeaf	Quality Manager	Green Acres	220,000	Good
PlentiSol	Research Director	Balmington	250,000	Fair
FaunaFlex	Project Manager	North Elk	200,000	High

As well as gathering as much specific information as possible about the job, there are a number of key general questions that you as a prospective job hunter should address:

**1. What are your choices?**

As a simplification, assume that the choices in this example are to accept a single job offer, from one of the companies.

**2. What are your goals?**

You may have one or more goals that need to be compared and evaluated, for example:

- Minimize your daily travel time
- Maximize your financial compensation
- Improve your quality of life

**3. What are the criteria for evaluating your choices against your goals?**

In the real-life job-hunting situation, you typically have your own personal evaluation criteria, based on your requirements and past experiences. The process of evaluating your choices is often intuitive. However, the evaluation process can include satisfying more formal, numeric conditions, such as the requirement for a particular minimum salary.

In the Oracle RTD decision process, evaluation criteria are implemented by an ordering algorithm that prioritizes choices by assigning scores to them.

The scores for each Oracle RTD choice are computed using one or more of the following scoring methods:

- The rule driven scoring method uses explicit business rules, such as "Salary must be at least 200000" or "Promotion Prospect must be Good or High"
- The model driven scoring method uses implicit rules discovered from analyzing historical data stored in an Oracle RTD predictive model

When there are several performance goals for a decision, you can weight the goals. For example:

Performance Goal	Weighting
Maximize your daily travel time	30
Maximize your financial compensation	50
Improve your quality of life	20

Oracle RTD can score each choice against each performance goal or weighted combination of goals.

The net effect is that Oracle RTD provides a numeric score for each choice, such as in the following table:

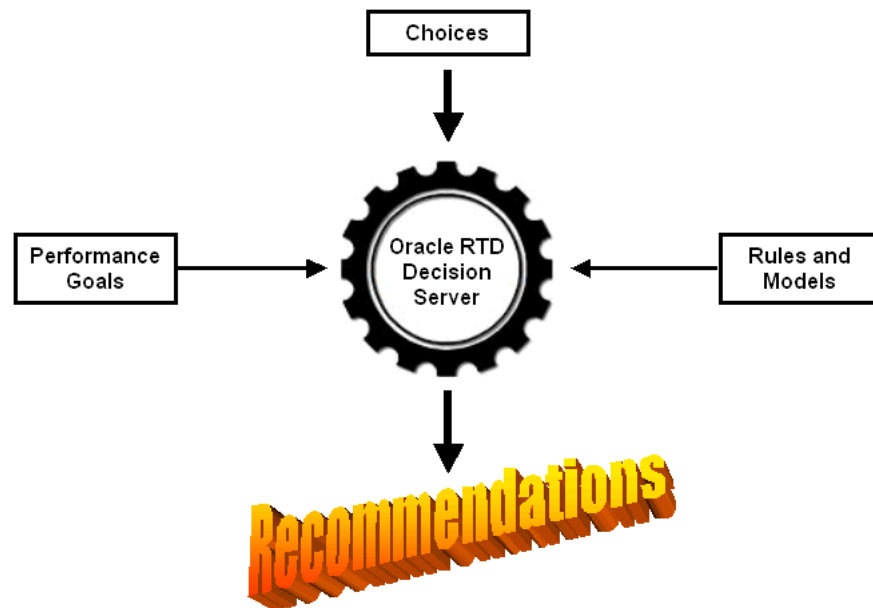
Company	Job	Location	Salary	Promotion Prospects	Choice	Score
VeriLeaf	Quality Manager	Green Acres	220,000	Good	Accept VeriLeaf job	60
PlentiSol	Research Director	Balmington	250,000	Fair	Accept PlentiSol job	50
FaunaFlex	Project Manager	North Elk	200,000	High	Accept FaunaFlex job	80

### Oracle RTD Decision Making Features Overview

The overall principles and underlying elements described in the job hunting example are incorporated as basic features of Oracle RTD, as shown in the following table:

Questions in the Decision Making Process	Oracle RTD Features
What are your choices?	Choices
What are your goals?	Performance Goals
What are your criteria for evaluating those goals?	Rules and Models

The following diagram shows a high-level overview of how these features interact to fulfill the basic objective of Oracle RTD, namely to provide recommendations from a number of alternatives or choices.



For more information on these features and how to use them in Oracle RTD, see the following sections:

- [Section 2.4, "The Oracle RTD Decision Process"](#)
- [Section 2.5, "More About the Oracle RTD Decision Process Elements"](#)

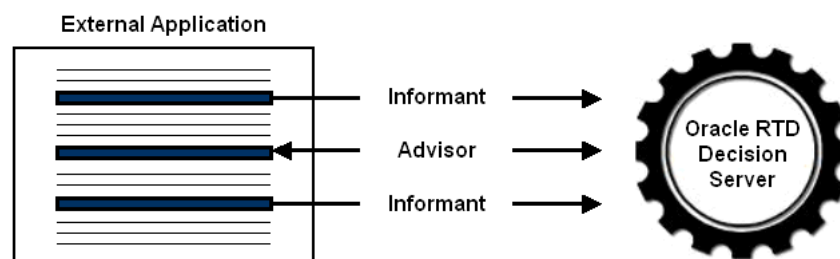
In general, Oracle RTD connects with other applications, and passes its recommendations to these external applications. See the next section for more information about how Oracle RTD integrates with external applications.

## 2.2 Integration Points Between External Applications and Oracle RTD

Applications that you develop to interact with Oracle RTD are referred to as external applications. Typically external applications consist of many processing steps and stages. The points at which external applications communicate with Oracle RTD are generically known as **Integration Points**.

There are two main types of Integration Point:

- An **Informant** is a process that passes data from the external application to Oracle RTD
- An **Advisor** is a two-way process, that both passes data from the external application to Oracle RTD, and also receives data sent back from Oracle RTD to the external application



Advisors are the main method by which an external application requests and receives recommendations from Oracle RTD.

Each external application can have many Informants and Advisors.

## 2.3 Overview of Oracle RTD Integration with External Applications

Many applications are based on a dialog with a user, which leads to the application presenting alternative strategies or choices to the user.

Typically, the dialog between the application and the user proceeds as follows:

1. The user starts a transaction.
2. The application retrieves information about the user.
3. Optionally, the user provides extra information concerning the transaction.
4. The application presents the user with one or more choices.
5. The user accepts or ignores the choice or choices.
6. Optionally, the previous two stages may be performed several times during the course of the transaction.
7. The user ends the transaction.

To determine which choices to present to a user, external applications can use various factors, such as:

- Profile information about the user
- Current information about the transaction
- The user's preferences, if known
- Past activities or transactions associated with the user
- User access method, such as the Web or a custom interface
- Time of day, month, or season

Oracle RTD provides a set of tools that can analyze all these factors, and recommend the best choices to the external applications. Through these recommendations, Oracle RTD enables the companies that run the external applications to make better business decisions.

Figure 2–1 shows, in outline form, how a typical application interfaces with Oracle RTD.

**Figure 2–1 Overview of External Application Integration with Oracle RTD**

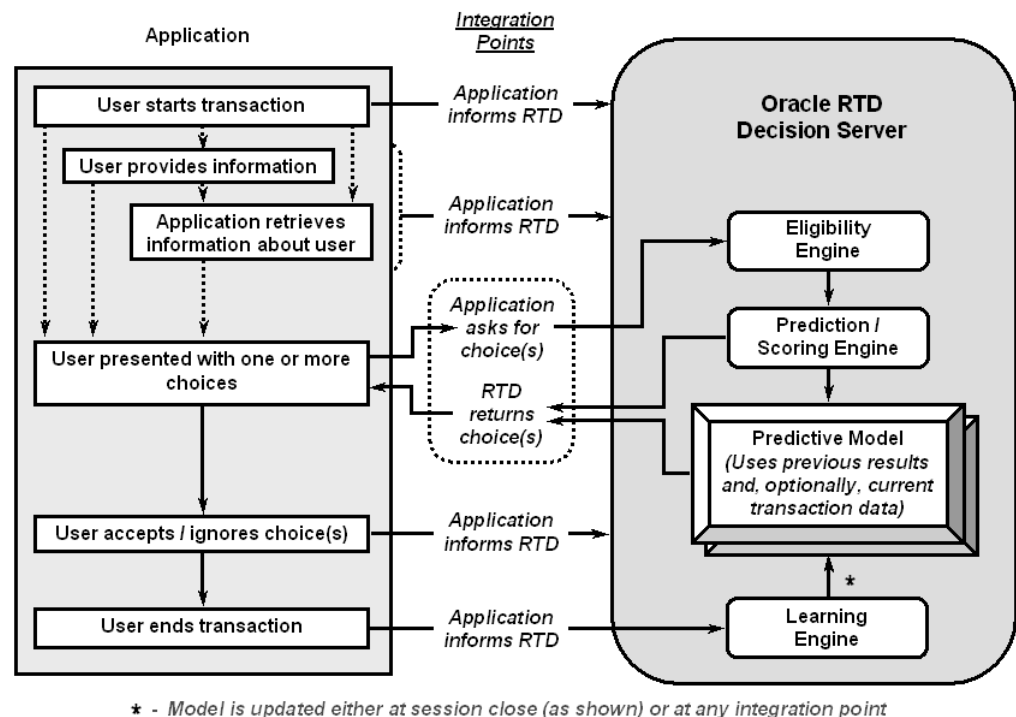


Figure 2–1 shows one Advisor and four Informants, the Informants corresponding to the following key stages in the application:

- The user has started the transaction.
- The external application has acquired more information about the user.
- The user has accepted or rejected a choice.
- The user has ended the transaction.

## 2.4 The Oracle RTD Decision Process

This section shows the general Oracle RTD decision process flow. For details about the Oracle RTD features and elements used in the process flow, see [Section 2.5, "More About the Oracle RTD Decision Process Elements."](#)

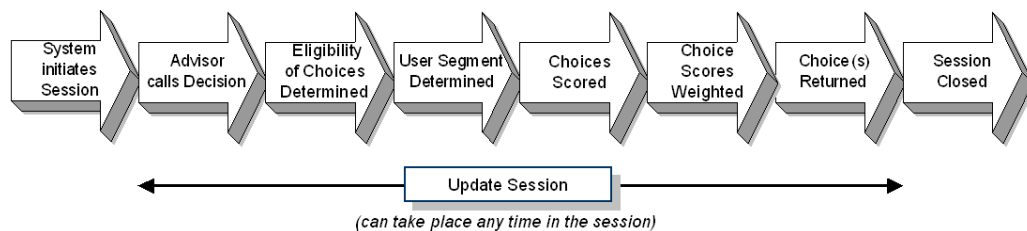
The Oracle RTD decision process is based on a framework that takes into account the following factors:

- The overall performance goals with which an organization is concerned
- The action required to score each of the available choices
- A weighting of the performance goals based on segments of the population

Decisions are called by Advisors to score Choices, and return one or more Choices from a Choice Group. The set up of a Decision must include at least one Choice Group from which Choices are selected, and a function or rule to score the Choices. At run time, the Decision collects all the eligible Choices that exist in each of the Choice Groups. Then, the Choices are scored to finally determine the ranked order to send back through the Advisor.

[Figure 2–2](#) shows the basic Oracle RTD processes, which include session start and finish, as well as the Oracle RTD decision process steps.

**Figure 2–2 Oracle RTD Decision Process Flow**



The steps represent the different stages in the overall process of acquiring the necessary data and processing a decision, as follows:

1. The system initiates the session.

When a user log on, and the external application connects to Oracle RTD, Oracle RTD establishes a session.

The external application generally acquires as much information about the user as possible, and passes it to Oracle RTD using one or more Informants.

Oracle RTD may also retrieve further information from Data Sources and Entities defined in the Inline Service associated with the external application.

2. The Advisor calls a Decision.

A request through an Advisor call initiates the decision process. The set of choices to evaluate for the decision is then determined for each of the associated Choice groups.

3. The eligibility of Choices is determined.

The eligibility rules for the Choices are invoked, to determine which Choices pass on to the next stage of the decision process.

4. The user segment is determined.

Filtering rules, if created, are then used to segment the user population. Based on the segment, the designated weightings for each of the Performance Goals is used in scoring each eligible Choice.

5. The Choices are scored.

All eligible Choices are scored for each associated Performance Goal.

6. The Choice scores are weighted.

Based on the segment, different weights are applied to the Performance Goal scores.

7. The Choice or Choices are returned to the external application.

Oracle RTD returns one or more Choices to the external application, passing Choice names and any designated Choice attribute that the external application needs. The requesting application then displays the Choices or processes the information accordingly.

8. The session information is updated.

This step can take place at any stage of the decision process. Its main effect is to update the Oracle RTD server with any new available information about the given session.

In addition, Models can be updated from the session information either at specified integration points or at the end of the session.

9. The session is closed.

The active Oracle RTD session is closed and any wrap up logic is executed, including learning on any Models defined to learn at session close.

## 2.5 More About the Oracle RTD Decision Process Elements

This section provides more details about the following Oracle RTD elements used in the decision processing framework:

- [Performance Goals](#)
- [Decisions](#)
- [Choice Groups and Choices](#)
- [Eligibility Rules](#)
- [Filtering Rules](#)
- [Scoring Rules](#)
- [Models](#)

### 2.5.1 Performance Goals

Designers creating a decision process for an organization must consider the overall Performance Goals of the organization. Performance Goals consist of the specific metrics with which the organization has chosen to measure its success. These goals are then associated to choice groups and decisions to identify how each choice will be scored against those them. Some common performance metrics are revenue, costs, number of products per customer, and customer satisfaction.

If you set more than one Performance Goal in an Inline Service, you must specify the relative importance of each one by assigning normalization factors for each Performance Goal.

## 2.5.2 Decisions

Decisions score eligible Choices and rank them based on the weightings given for associated Performance Goals.

Oracle RTD supports the following types of Decisions:

- Rule-driven Decisions
- Model-driven Decisions
- Hybrid Decisions

Rule-driven Decisions are defined in business related terms expressed by business users. An example could be the business rule not to sell credit cards to customers when their credit rating is low.

Model-driven Decisions are driven by scores calculated and determined by Models formed from empirical data. An example could be the decision to present an Overdraft Protection offer to a call center user who lives in California, whose occupation is graphical artist, and who has called to change his address. Based on its previous learnings, the model has determined that similar users are 61% likely to accept the Overdraft Protection offer.

Hybrid Decisions use the scoring methods of both the Rule-driven and the Model-driven decisions.

In general, each Decision may be associated with:

- One or more Choice Groups
- One or more Performance Goals
- One or more segments of the user population, where each segment can have different weightings for each Performance Goal

## 2.5.3 Choice Groups and Choices

Choice Groups and Choices are the Inline Service elements from which Decisions draw their possible Choices, and which become targets of analysis for Choice and Choice Event Models.

Choice Groups and Choices form a hierarchy, where:

- Each Choice belongs to one Choice Group only
- Each Choice Group can have one or more sub-Choice Groups

Choices exist only at the lowest level of a Choice Group hierarchy branch.

Choices can be used by a Decision, so that they can be returned by Advisors, and can be registered to either Choice or Choice Event models through Informants.

### Choice Group and Choice Attributes

Choice Groups and Choices have attributes, that is, data used in the processing and presentation of Choices.

Typically, you define the attributes of Choices at a higher Choice Group level, where you can also specify default values for the attributes. The Choice Group attributes are



inherited by lower level Choice Groups and Choices. You can override default values at the lower levels.

### **Static and Dynamic Choices**

Choices can either be static or dynamic.

Static Choices are explicitly defined in the Inline Service.

Dynamic Choices are Choices that are stored and maintained in an external application, such as promotions stored in a separate marketing application. When required for the decision process, Dynamic Choices are retrieved from the external application. The mechanisms for retrieving and using Dynamic Choices are defined in the Inline Service, but the actual Dynamic Choice values may vary for each user session.

For more information about Dynamic Choices, see *Oracle Real-Time Decisions New Features Guide Version 2.2.1*.

## **2.5.4 Eligibility Rules**

Choices and choice groups have rules that determine their eligibility to participate in a decision.

You can define eligibility rules at the Choice Group and Choice levels.

Choices inherit rules from higher levels, and may also have their own rules. At each level, a logical AND is performed between the higher-level rules and the current-level rule, with the result placed in the current-level element.

## **2.5.5 Filtering Rules**

Choices and Choice Groups can use filtering rules as another form of eligibility. In addition, a filtering rule can also be used to segment the user population for which Decisions are being made, and controls the effect of each Performance Goal associated to the Decision.

## **2.5.6 Scoring Rules**

Scoring rules are similar in setup to eligibility rules, but rather than evaluate the rule to a TRUE/FALSE outcome, a numeric score is returned instead. A score can be computed for a given Performance Goal tied to a Choice, and can affect the rank of the Choices in the decision process.

## **2.5.7 Models**

There are two standard types of model in Oracle RTD:

- Choice Model
- Choice Event Model

Each Choice Model or Choice Event Model is always associated with a single Choice Group.

Both types of model can be used for prediction and for generating analysis reports.

### 2.5.7.1 Model Usage Summary

#### Input

The main objective of any model is to show, for each choice of the associated choice group, what factors influenced a particular choice.

Models are updated with, and "learn" from the following data:

- All the data in the Session entity - specifically, all the Session attributes, unless specifically excluded from analysis through the Inline Service configuration

In addition, Choice Event Models also require event-related details. For more information, see [Section 2.5.7.3, "Choice Event Models."](#)

The update and learning process happens in a transaction either at session close or at any integration point.

#### Outputs

Both types of model can be used for prediction and for generating analysis reports.

The outputs generated directly and indirectly from a model are as follows:

- Model scores for a given Choice which can be used as part of the decision process
- Analytic reports in the Decision Center
- Model snapshots that enable users to generate their own analytic reports

### 2.5.7.2 Choice Models

The main objective of a Choice Model is, for each choice, to derive meaningful information from the data associated simply with the choice itself. A Choice Model does not need the extra dimension of base and positive outcome events, which are required for Choice Event Models.

For instance, in a call center application, one of the key data elements is the reason for a call. After collecting more information about the call and the caller, you can provide this information to a Call Reason Choice Model, and then use this in Oracle RTD Decision Center to analyze and compare the driving attributes of different call reasons.

Another example of a Choice Model is an Abandonment Model, with two choices, Abandoned and Not Abandoned. For both choices, the model stores data associated with the user and the transaction, and whether the user abandoned the transaction before completion. You can use the model not only to analyze potential abandonment factors, but also to predict the likelihood of whether subsequent users will abandon their transactions.

### 2.5.7.3 Choice Event Models

For each Oracle RTD Choice Event Model, in addition to specifying a Choice Group, you must also specify one Base event and one or more Positive Outcome events.

In the simplest case, there are two significant events in a transaction, the presentation of a choice and the acceptance of the choice.

In Oracle RTD, events are defined at the Choice Group level, and selected within the Model to describe "base" and "success" parameters.

For each Choice Event Model, you must define:

- One Base event, used as the base for analysis.

Typically, this is the event associated with the presentation of the choice.

- One or more Positive Outcome events, each of which indicate a successful prediction.

Typically, the standard positive outcome event is the event associated with the acceptance of the choice.

## 2.6 General Oracle RTD Elements and Features

Some Oracle RTD objects have a general usage within and across Oracle RTD processes. This section describes the following general Oracle RTD elements and features:

- [Data Sources](#)
- [Entities](#)
- [Functions](#)

### 2.6.1 Data Sources

A Data Source is configured in an Inline Service to access data from an outside source. The structure and format of Data Sources can vary, as follows:

- The rows and columns of a database table or view
- The output values and result row sets from a stored procedure

A Data Source can be configured to retrieve either a single record or multiple rows.

Each Data Source contains Input and Output columns:

- The Input columns are used in the WHERE clause of the query to the Data Source to select the rows to retrieve.
- The Output columns are the data that is retrieved and used by Oracle RTD in the decision process.

### 2.6.2 Entities

An Entity is a logical representation of data, that can be populated from one or more Data Sources, through data retrieved by an Integration Point, or through functional derivations. Entities are the data objects that can be used by the other Oracle RTD elements, and form a logical level of abstraction from Data Sources and Integration Points.

An Entity is a set of named attributes and methods to access the attributes. One attribute per Entity is usually designated as the Entity key.

An attribute of an Entity is analogous to a column of a database table, with one important distinction: an Oracle RTD attribute may consist either of one value or many values. The type of attribute that can have multiple values is called an Array attribute.

The integration of Entities and their component attributes to the appropriate data is implemented by mapping. You can explicitly map Entity attributes to Data Source columns, or you can implicitly map them through the use of Java functions that populate the Entity attributes.

An Entity, while it contains its own attributes, may also be an attribute of another Entity. For example, a customer can have many orders. In Oracle RTD, you can define Customers and Orders as separate Entities, mapped from corresponding Data Sources. You can then specify the Orders Entity to be an attribute of the Customers Entity.

### **Session Entities**

**The Session is the fundamental Oracle RTD unit of run-time data. Data is kept in memory for the duration of the Session. Every Inline Service contains one Session Entity.**

For a Model to be able to learn from the attributes of a non-Session Entity, that Entity must be defined as an attribute of the Session Entity.

For example, in an Inline Service, you can define Customer, Call, and Product as logical Entities, and then add these as attributes to the Session Entity, so that the Oracle RTD server can use these Entities as inputs to the Models.

## **2.6.3 Functions**

Functions, written in Java, provide extra processing capabilities to many Oracle RTD elements. For example, selection functions can be used by decisions as a custom way to make a choice.

Functions can also serve as general-purpose code, for example, to determine date differences, or to convert data into different data types.

Other users of functions include:

- Populating derived entity attributes
- Comparing values in choice eligibility rules
- Retrieving lookup values
- Writing to log files

Functions may also call other functions.

## **2.7 Oracle RTD Inline Services**

An Oracle RTD Inline Service consists of all the Oracle RTD elements necessary to interface with an external application and model the desired business process.

The main elements of an Inline Service are the following:

- Data Sources
- Entities
- Integration Points
- Choice Groups and Choices
- Decisions
- Filtering Rules
- Scoring Rules
- Models
- Performance Goals
- Functions

Not all Inline Services have all of these elements. The specific requirements of each external application determine which elements are needed in the associated Inline Service.

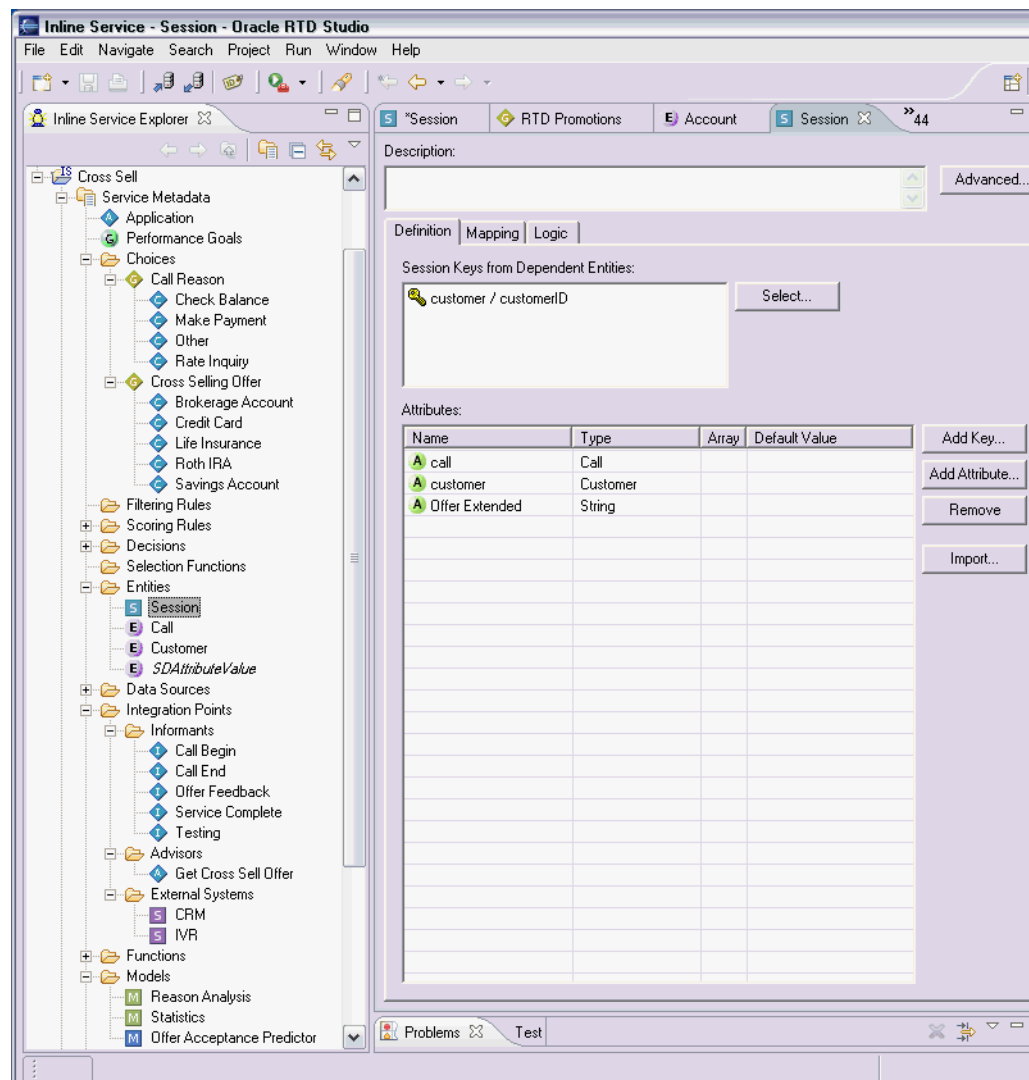
For more information on how to configure an Inline Service, see the [Related Documents](#) in the [Preface](#) chapter, in particular the *Oracle Real-Time Decisions Decision Studio Reference Guide*.

## 2.8 Introduction to Oracle RTD Decision Studio

In Oracle RTD, you define the Inline Service elements in the Oracle RTD Decision Studio. You must first configure, then deploy an Inline Service to the Oracle RTD server before the Inline Service can be used by an application.

[Figure 2–3](#) shows some of the elements of an Inline Service, called Cross Sell, as displayed in the Decision Studio.

**Figure 2–3 Example of an Inline Service Displayed in Decision Studio**



For more information about how to define and deploy Inline Services, see *Oracle Real-Time Decisions Decision Studio Reference Guide*.

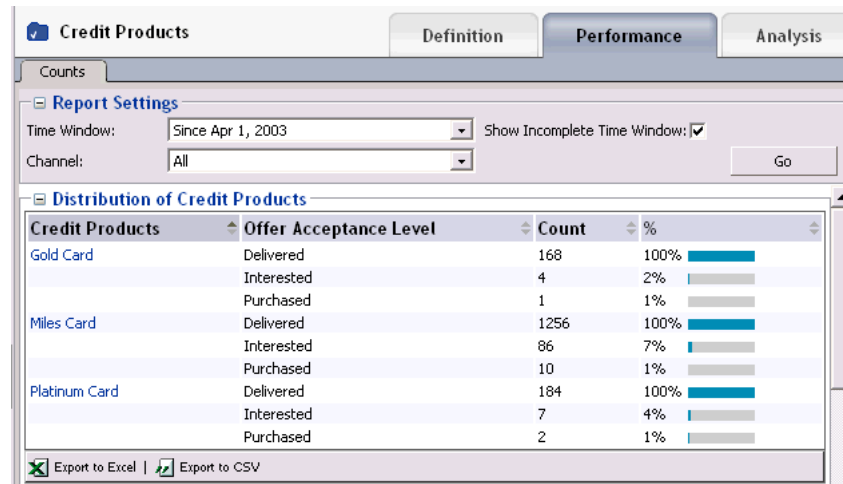
## 2.9 Analytic Reports and the Oracle RTD Decision Center

Oracle RTD Decision Center is a client tool for business users to explore, analyze, examine, and even modify the structure and data gathered by a deployed Inline Service.

The Oracle RTD Decision Center provides a variety of analytic reports, both for performance analysis and model analysis.

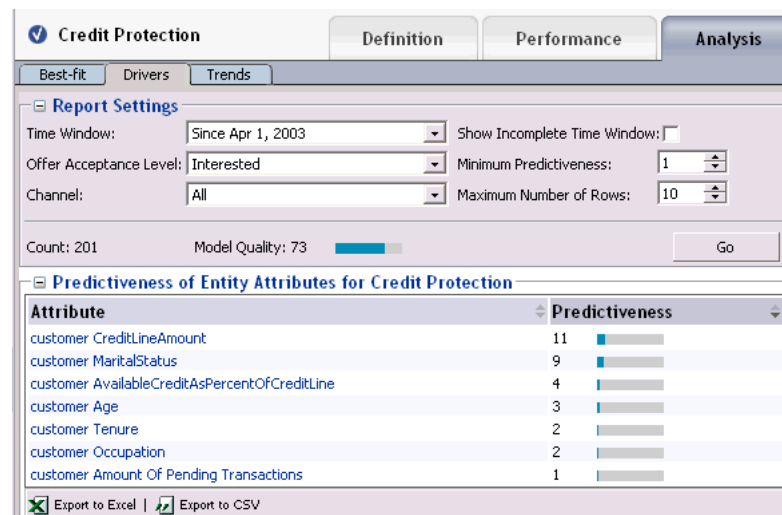
For example, there are several reports at the choice group and choice level, such as the following examples from a Cross Sell application:

### ■ Choice Group Performance Counts



The Choice Group Performance Counts shows the total counts for each choice or choice event occurrence in a choice group.

### ■ Choice Analysis Drivers

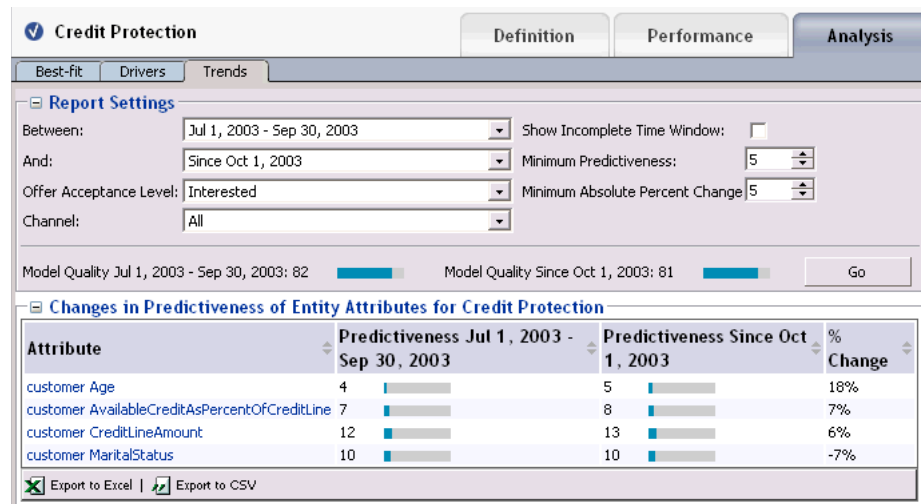


The Choice Analysis Drivers report identifies the attributes that are influential as drivers of predictiveness for each of the choices.

Predictiveness is a measure of the relationship strength between entity attributes, that are the model input, and choice and choice events, that are the model output.

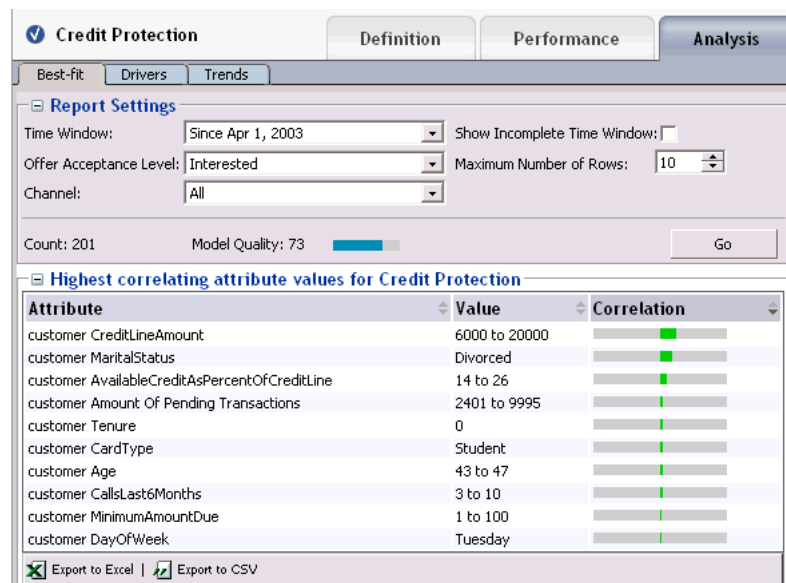
A drilldown on any of the attribute hyperlinks will reveal additional reports about the attribute values themselves.

### ■ Choice Analysis Trends



The Choice Analysis Trends report shows the change of predictiveness for each of the attributes for a choice over two selected model time windows.

### ■ Choice Analysis Best Fit



The Choice Analysis Best Fit report shows all the attributes and values that are most likely to predict the specified event outcome.

Oracle RTD also provides a variety of reports that show the effectiveness of entities and entity attributes for predicting choices.

For more information about how to view, analyze, and modify the structure and data of Inline Services in the Decision Center, see *Oracle Real-Time Decisions Decision Center User Guide*.





---

## RTD Siebel ECommerce Inline Service Elements

This chapter describes the elements in the RTD Siebel ECommerce Inline Service. It contains the following topics:

- [Section 3.1, "Overview of Oracle RTD Integration with Siebel E-Commerce"](#)
- [Section 3.2, "Data Sources"](#)
- [Section 3.3, "Entities"](#)
- [Section 3.4, "Informants"](#)
- [Section 3.5, "Advisors"](#)
- [Section 3.6, "Choice Groups and Choices"](#)
- [Section 3.7, "Decisions"](#)
- [Section 3.8, "Models"](#)
- [Section 3.9, "Functions"](#)

### 3.1 Overview of Oracle RTD Integration with Siebel E-Commerce

[Figure 3–1](#) shows the high-level integration points between the Siebel E-Commerce application and Oracle RTD.

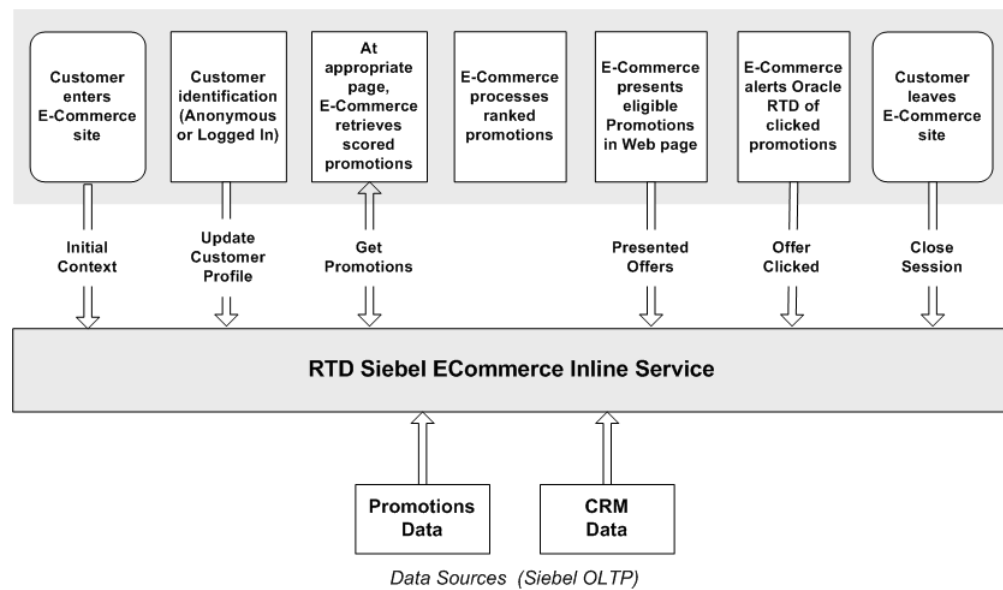
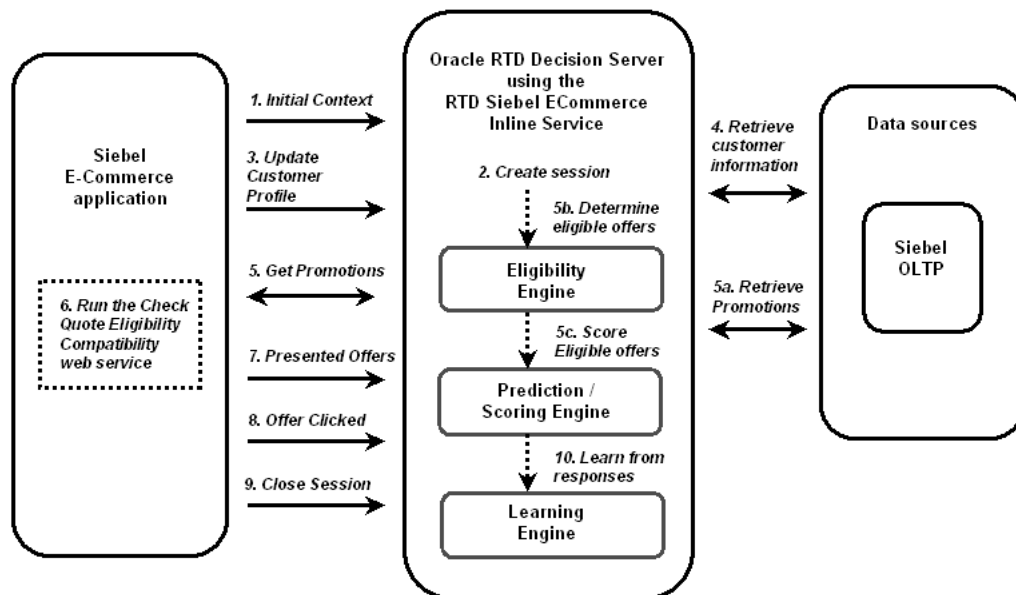
**Figure 3–1 High-Level Siebel E-Commerce Integration Flow with Oracle RTD**

Figure 3–2 shows a detailed integration flow between the Siebel E-Commerce application and the Inline Service RTD Siebel ECommerce.

**Figure 3–2 Detailed Siebel E-Commerce Integration Flow with Oracle RTD Inline Service**

1. The user logs in to the Siebel E-Commerce application.

The Siebel E-Commerce application initiates the context for an Oracle RTD session. It creates a session key (timestamp and random number combination), and sends the session key to the RTD Siebel ECommerce Inline Service through the Initial Context Informant.

2. Oracle RTD creates a session.

3. If the user is a registered customer, the Siebel E-Commerce application sends customer account and contact information to Oracle RTD through the Update Customer Profile Informant.
4. Oracle RTD extracts further customer information for the registered users from the Data Sources defined in the Inline Service.
5. The Siebel E-Commerce application sends the Advisor Get Promotions to Oracle RTD, specifying catalog and category id as parameters, if these were not already sent with a prior Informant.
  - a. Oracle RTD extracts the promotions from the Data Sources defined in the Inline Service.
  - b. The promotions choices are passed to the Oracle RTD Eligibility Engine.
  - c. The eligible promotion choices are scored in accordance with the performance goals associated to this decision and then ranked.

The default scoring process for each choice is based on the likelihood score returned from the Promotions Acceptance model defined in the RTD Siebel ECommerce Inline Service for the performance goal, Maximize Acceptance Likelihood.

At the final stage of Get Promotions Advisor processing, Oracle RTD returns promotion choices to the Siebel E-Commerce application.

6. The Siebel E-Commerce application runs the returned promotions through Siebel's CheckQuoteEligibilityCompatibility web service before displaying the final set of eligible promotions as Hot Offers to the user.
7. The Siebel E-Commerce application sends Oracle RTD details of the promotions presented to the user through the Presented Offers Informant.
8. The Siebel E-Commerce application sends Oracle RTD details of the promotions that the user clicked through, if any, through the Offer Clicked Informant.
9. When the user logs out from the Siebel E-Commerce application, the Siebel E-Commerce application informs Oracle RTD through the Close Session Informant.

For anonymous users, the Oracle RTD session will timeout, closing the session automatically.

10. Oracle RTD applies the feedback from the web session to update the Promotions Acceptance Model, to refine and improve the model for future predictions.

## 3.2 Data Sources

As released, the RTD Siebel ECommerce Inline Service has been configured to extract various forms of customer data after a customer has been identified. Oracle RTD retrieves the data into Oracle RTD data sources using JDBC to the Siebel OLTP. These data sources can be categorized into the following areas:

- Customer Profile - Information related to the customer on the E-Commerce site
- Assets - Information related to a customer's current product holding
- Past Web Interactions - Information related to past Oracle RTD web sessions that a given customer has had
- Catalog Structure - Category/Catalog information directly related to the page that the customer is on in the E-Commerce site

- Promotions - Source data that will be used with Oracle RTD's dynamic choices for presenting the best promotion to the E-Commerce customer

This section contains the following topics:

- [Section 3.2.1, "Accounts DS"](#)
- [Section 3.2.2, "Address DS"](#)
- [Section 3.2.3, "Assets DS \(Multiple Rows\)"](#)
- [Section 3.2.4, "Catalog Category DS"](#)
- [Section 3.2.5, "Catalog DS"](#)
- [Section 3.2.6, "Contact Address DS"](#)
- [Section 3.2.7, "Contact DS"](#)
- [Section 3.2.8, "Image DS"](#)
- [Section 3.2.9, "Industry DS"](#)
- [Section 3.2.10, "List of Values DS"](#)
- [Section 3.2.11, "Product Line DS"](#)
- [Section 3.2.12, "Products DS"](#)
- [Section 3.2.13, "Promo Category Catalog DS"](#)
- [Section 3.2.14, "Promotions DS"](#)
- [Section 3.2.15, "Web History DS"](#)

### 3.2.1 Accounts DS

The Account data source extracts, from the Siebel data source S\_ORG\_EXT, account based information including the keys to additional account tables that store other profile information, for example, Address information.

**Table 3–1 Account DS - SIEBEL\_S\_ORG\_EXT**

Parameter	Column Names	Comments
Output	PR_ADDR_ID CUST_STAT_CD OU_TYPE_CD PR_INDUST_ID	None
Input	ROW_ID	Populated by Customer Id passed through from front end, and set for the session.

### 3.2.2 Address DS

The Address data source extracts, from the Siebel data source S\_ADDR\_ORG, address information tied to the customer account.

**Table 3–2 Address DS - SIEBEL.S\_ADDR\_ORG**

Parameter	Column Names	Comments
Output	COUNTRY STATE ZIPCODE	None

**Table 3–2 (Cont.) Address DS - SIEBEL.S\_ADDR\_ORG**

Parameter	Column Names	Comments
Input	ROW_ID	Populated by PR_ADDR_ID from the S_ORG_EXT data source.

### 3.2.3 Assets DS (Multiple Rows)

The Assets data source extracts, from the Siebel data source S\_ASSETS, asset based information tied to the customer account.

**Table 3–3 Assets DS - SIEBEL.S\_ASSETS**

Parameter	Column Names	Comments
Output	CREATED PROD_ID	None
Input	OWNER_ACCNT_ID	Populated by Customer Id passed through from front end, and set for the session.

### 3.2.4 Catalog Category DS

The Catalog Category data source extracts, from the Siebel data source S\_CTLG\_CAT, information related to the category that the user has drilled down into from the web front end. The Category Id will be sent through the Oracle RTD integration with the web front end.

**Table 3–4 Catalog Category DS - SIEBEL.S\_CTLG\_CAT**

Parameter	Column Names	Comments
Output	CTLG_ID NAME PAR_CAT_ID	None
Input	ROW_ID	Populated by the Category Id passed in through the Informant Get Promotions.

### 3.2.5 Catalog DS

The Catalog data source extracts, from the Siebel data source S\_CTLG, information related to the Catalog that the user has drilled down into from the web front end. The Catalog Id will be sent through the Oracle RTD integration with the web front end.

**Table 3–5 Catalog DS - SIEBEL.S\_CTLG**

Parameter	Column Names	Comments
Output	NAME	None
Input	ROW_ID	Populated by the Catalog Id passed in through the Informant Get Promotions.

### 3.2.6 Contact Address DS

The Contact Address data source extracts, from the Siebel data source S\_ADDR\_PER, address information tied to the customer contact.

**Table 3–6 Contact Address DS - SIEBEL.S\_ADDR\_PER**

Parameter	Column Names	Comments
Output	COUNTRY STATE ZIPCODE	None
Input	ROW_ID	Populated by PR_CON_ADDR_ID from the S_CONTACT data source.

### 3.2.7 Contact DS

The Contact data source extracts, from the Siebel data source S\_CONTACT, contact based information tied to the person that is using the web front end.

**Table 3–7 Contact DS - SIEBEL.S\_CONTACT**

Parameter	Column Names	Comments
Output	PR_ACT_ID PR_CON_ADDR_ID PR_OU_ADDR_ID	None
Input	ROW_ID	Populated by the Contact Id passed in through the Informant Update Customer.

### 3.2.8 Image DS

The Image data source extracts, from the Siebel data source S\_LIT, image name related information based on the image id tied to a promotion record.

**Table 3–8 Image DS - SIEBEL.S\_LIT**

Parameter	Column Names	Comments
Output	FILE_EXT FILE_NAME NAME ROW_ID	None
Input	ROW_ID	Populated by the Image Id value tied to each promotion record.

### 3.2.9 Industry DS

The Industry data source extracts, from the Siebel data source S\_INDUST, primary industry information tied to the account associated to the person that is using the web front end.

**Table 3–9 Industry DS - SIEBEL.S\_INDUST**

Parameter	Column Names	Comments
Output	NAME	None
Input	ROW_ID	Populated by the Industry Id tied to the Account record.

### 3.2.10 List of Values DS

The List of Values data source extracts, from the Siebel data source S\_LST\_OF\_VAL, lists of values for any column linked to a list of values id. This can be applied to the Account table, Contact table, and so on.

**Table 3–10 List of Values DS - SIEBEL.S\_LST\_OF\_VAL**

Parameter	Column Names	Comments
Output	NAME	None
Input	ROW_ID	Populated by various inputs that may map to this table.

### 3.2.11 Product Line DS

The Product Line data source extracts, from the Siebel data source S\_PROD\_LN, product line information tied to products extracted from S\_PROD\_INT.

**Table 3–11 Product Line DS - SIEBEL.S\_PROD\_LN**

Parameter	Column Names	Comments
Output	NAME	None
Input	ROW_ID	Populated by the Product Line id value tied to each product record.

### 3.2.12 Products DS

The Products data source extracts, from the Siebel data source S\_PROD\_INT, product based information tied to records extracted from the Assets data source.

**Table 3–12 Products DS - SIEBEL.S\_PROD\_INT**

Parameter	Column Names	Comments
Output	DESC_TEXT INCLSV_ELIG_RL_FLAG NAME PR_PROD_LN_ID	None
Input	ROW_ID	Populated by the Product Id tied to each Asset record.

### 3.2.13 Promo Category Catalog DS

The Promo Category Catalog data source extracts, from the Siebel data source S\_CTLG\_CAT\_PROD, information related to the categories and catalogs with which a promotion has been associated. This data can be used for future eligibility rules if necessary to show only promotions tied to a specific category or catalog. Category and Catalog information associated to promotions is calculated through the Set Category String and Set Catalog String functions.

**Table 3–13 Promo Category Catalog DS - SIEBEL.S\_CTLG\_CAT\_PROD**

Parameter	Column Names	Comments
Output	CTLG_CAT_ID	None
Input	ROW_ID	Populated by each promotion Row Id.

### 3.2.14 Promotions DS

The Promotions data source extracts, from the Siebel data source S\_PROD\_INT, promotion data used to be populated in the Dynamic Choice Group, Siebel Promotions. As a result, these records are evaluated in decisioning and used by the Advisor Get Promotions.

**Table 3–14 Promotions DS - SIEBEL.S\_PROD\_INT**

Parameter	Column Names	Comments
Output	DESC_TEXT INCLSV_ELIG_RL_FLG NAME PROD_IMAGE_ID ROW_ID	None
Input	ORDERABLE_FLG PROMO_TYPE_CD	Both inputs coded in the Promotions List entity.

### 3.2.15 Web History DS

This data source, a custom table for the E-Commerce solution, resides in the SDDS database and is used to capture web history data for customers that have provided their account id. The data in the table is then evaluated through the Web Interaction entity. Create Table scripts are provided with this Inline Service for this table and can be found in the **etc** folder of the Inline Service directory.

**Table 3–15 Web History DS - SDDS.RTD\_WEB\_HISTORY**

Parameter	Column Names	Comments
Output	CATALOGS_VISITED CATEGORIES_VISITED CREATED OFFER_CLICKED SESSION_ID	None
Input	ACCOUNT_ID	Populated by the Customer Id provided by the Informant Update Customer Profile.

## 3.3 Entities

Oracle RTD entities can be broken down into the following categories:

- **Session**

This special category contains the [Session Entity](#) only. Every Oracle RTD Inline Service has a Session entity, which defines the entities to be associated with the session.

- **Customer Profile**

The Customer Profile entities build out what is known about the customer utilizing the web front end. This category contains the following entities:

- [Account Entity](#), [Assets Entity](#), [Contact Entity](#),  
[Industry Entity](#), [Products Entity](#), [Raw Assets Entity](#)

- **Web Interaction**



The Web Interaction entities are used to capture information related to the web interaction, such as which Catalog and Category the user is in on the web front. This category contains the following entities:

- [Raw Web Interactions Entity](#), [Web Interaction Entity](#)

#### ■ **Product Catalog/Category**

These entities are used to map product catalog and category information received from the front end UI. This category contains the following entities:

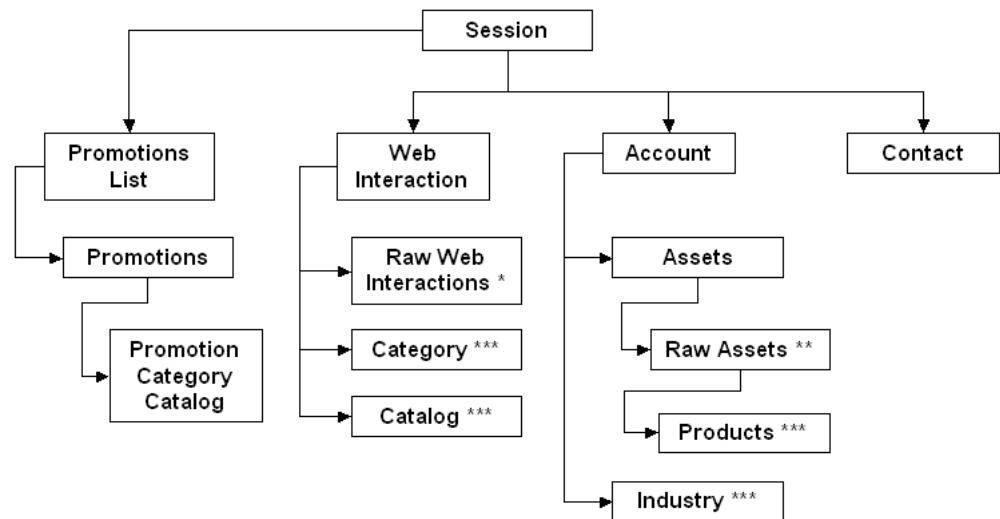
- [Catalog Entity](#), [Category Entity](#)

#### ■ **Promotions**

The Promotions entities are used for building out dynamic choice, which are utilized by the Advisor Get Promotions. These entities are not used for modeling. This category contains the following entities:

- [Promo Category Catalog Entity](#), [Promotions Entity](#), [Promotions List Entity](#)

The following diagram shows the overall logical entity model of the RTD Siebel ECommerce Inline Service.



• Raw Web Interactions represented as array of Past Web Interactions

\*\* Raw Assets represented as array of Assets

\*\*\* Entity values populated through Lookup functions

The rest of this section contains the following topics:

- [Section 3.3.1, "Session Entity"](#)
- [Section 3.3.2, "Account Entity"](#)
- [Section 3.3.3, "Assets Entity"](#)
- [Section 3.3.4, "Catalog Entity"](#)
- [Section 3.3.5, "Category Entity"](#)
- [Section 3.3.6, "Contact Entity"](#)
- [Section 3.3.7, "Industry Entity"](#)

- [Section 3.3.8, "Products Entity"](#)
- [Section 3.3.9, "Promo Category Catalog Entity"](#)
- [Section 3.3.10, "Promotions Entity"](#)
- [Section 3.3.11, "Promotions List Entity"](#)
- [Section 3.3.12, "Raw Assets Entity"](#)
- [Section 3.3.13, "Raw Web Interactions Entity"](#)
- [Section 3.3.14, "Web Interaction Entity"](#)

### 3.3.1 Session Entity

The Session entity defines which entities should be included as part of the session. In addition to attributes configured directly in the session, entities included as session attributes are used as inputs to the Oracle RTD models.

The exceptions to this are entity attributes that have been specifically excluded for analysis but are used for other areas of the Inline Service. Examples of this are Id attributes that are used to join different data sources.

The Session entity has its session key set by the web front end to a unique value that represents a unique customer session.

**Table 3–16 Session Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Web Session Id	No	String	None	No	No	NA
Account	No	Account	None	Yes	Yes	NA
Contact	No	Contact	None	Yes	Yes	NA
Promotions List	No	Promotions List	None	No	No	NA
Session Date	No	Date	None	Yes	Yes	Default Value = System Date (calculated from the Return System Date() function)
Web Interaction	No	Web Interaction	None	Yes	Yes	NA

#### 3.3.1.1 Writing Output to RTD\_WEB\_HISTORY Table During Session Cleanup

Within the Cleanup Logic section of the Session Entity, the necessary code to write a record of the RTD Session to the RTD\_WEB\_HISTORY table has been included. This code, in conjunction with the LogWriter() and InsertMessageResult() functions, define how the columns in RTD\_WEB\_HISTORY are populated.

This code is only enabled, when the Application parameter **Enable RTD Web History** is set to True. Otherwise, no information will be populated to this table.

In the released version of the RTD Siebel ECommerce Inline Service, the Web Interaction attribute Past Web Interactions is populated through the records found in this table for a given Account Id.

### 3.3.2 Account Entity

**Table 3–17 Account Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Assets	No	Assets	NA	Yes	Yes	Placed in the Account entity to group customer owned assets
Country	No	String	Address DS.COUNTRY	Yes	Yes	Value joined through Primary Address Id
Industry	No	String	Industry Lookup (PR_INDUST_ID)	Yes	Yes	Value populated through the Industry lookup function which returns the Name value of the record on S_INDUST for the corresponding PR_INDUST_ID
Primary Address Id	No	String	Account DS.PR_ADDR_ID	No	No	NA
State	No	String	Address DS.STATE	Yes	Yes	Value joined through Primary Address Id
Status	No	String	Account DS.CUST_STAT_CD	Yes	Yes	NA
Type	No	String	Account DS.OU_TYPE_CD	Yes	Yes	NA
Zipcode	No	String	Address DS.ZIPCODE	Yes	Yes	Value joined through Primary Address Id

**Table 3–18 Data Source Input Values for Account Entity**

Data Source	Input Column	Type	Input Value
Account DS	ROW_ID	String	Session.Web Interaction.Account Id
Address DS	ROW_ID	String	Primary Address Id

### 3.3.3 Assets Entity

**Table 3–19 Assets Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Days Since Last Asset	No	Integer	None	Yes	Yes	Derived attribute populated through the ProcessAssets() function that gives the days since the last asset was created
Last Created Product	No	String	None	Yes	Yes	Derived attribute populated through the ProcessAssets() function that gives the name of the last created product
Last Created Product Line	No	String	None	Yes	Yes	Derived attribute populated through the ProcessAssets() function that gives the product line of the last created product

**Table 3–19 (Cont.) Assets Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Product Line	Yes	String	None	Yes	Yes	Derived array attribute populated through the ProcessAssets() function that provides a distinct list of product lines owned by the account
Product Name	Yes	String	None	Yes	Yes	Derived array attribute populated through the ProcessAssets() function that provides a distinct list of product names owned by the account
Assets	Yes	Raw Assets Entity	(See attributes below)	No	No	Array List of Assets extracted from the Raw Assets Entity (see Assets attributes below for breakdown)
Assets.Create Date	No	Date	Assets DS.CREATED	No	No	NA
Assets.Product Id	No	String	Assets DS.PROD_ID	No	No	NA
Assets.Product Line	No	String	None	No	No	Mapped on the Raw Asset Entity
Assets.Product Line Id	No	String	None	No	No	Mapped on the Raw Assets Entity
Assets.Product Name	No	String	Assets DS.PROD_ID	No	No	Actual name value for the asset product is extracted through a Product Lookup function which returns the Name of the product associated to the product id (The name value is on the table S_PROD_INT from the Products Entity)

**Table 3–20 Data Source Input Values for Assets Entity**

Data Source	Input Column	Type	Input Value
Assets DS	OWNER_ACCNT_ID	String	Session.Web Interaction.Account Id

### 3.3.4 Catalog Entity

**Table 3–21 Catalog Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Row Id	No	String	None	No	No	NA
Catalog Name	No	String	Catalog DS.NAME	No	No	Used to extract the Catalog name value of any given Catalog Id (Catalog Lookup function)

**Table 3–22 Data Source Input Values for Catalog Entity**

Data Source	Input Column	Type	Input Value
Catalog DS	ROW_ID	String	Row Id

---

**Note:** The Catalog Entity has caching enabled.

---

### 3.3.5 Category Entity

**Table 3–23 Category Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Row Id	No	String	None	No	No	NA
Catalog Id	No	String	Catalog Category DS. CTLD_ID	No	No	Used to extract the Catalog Id value of any given Category Id (Catalog Id Lookup function)
Category Name	No	String	Catalog Category DS. NAME	No	No	Used to extract the Category Name value of any given Category Id (Catalog Lookup function)
Parent Category Id	No	String	Catalog Category DS. PAR_CAT_ID	No	No	Used to extract the Parent Category Id value of any given Category Id (Parent Category Lookup function)

**Table 3–24 Data Source Input Values for Category Entity**

Data Source	Input Column	Type	Input Value
Catalog DS	ROW_ID	String	Row Id

---

**Note:** The Category Entity has caching enabled.

---

### 3.3.6 Contact Entity

**Table 3–25 Contact Entity**

Primary Address Id	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Country	No	String	Address DS.COUNTRY	Yes	Yes	Value joined through Primary Address Id
Primary Address Id	No	String	Contact DS.PR_OU_ADDR_ID	No	No	NA
State	No	String	Address DS.STATE	Yes	Yes	Value joined through Primary Address Id
Zipcode	No	String	Address DS.ZIPCODE	Yes	Yes	Value joined through Primary Address Id

**Table 3–26 Data Source Input Values for Contact Entity**

Data Source	Input Column	Type	Input Value
Address DS	ROW_ID	String	Primary Address Id
Contact DS	ROW_ID	String	Session.Web Interaction.Contact Id

### 3.3.7 Industry Entity

**Table 3–27 Industry Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Row Id	No	String	None	No	No	NA
Name	No	String	Industry DS.NAME	No	No	Used to extract the Industry name value tied to the Primary Industry id of the account (Industry Lookup function)

**Table 3–28 Data Source Input Values for Industry Entity**

Data Source	Input Column	Type	Input Value
Industry DS	ROW_ID	String	Row Id

---

**Note:** The Industry Entity has caching enabled.

---

### 3.3.8 Products Entity

**Table 3–29 Products Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Row Id	No	String	None	No	No	NA
Name	No	String	Products DS. NAME	No	No	Used to extract the product name value of assets tied to customers (Product Lookup function)
Product Line	No	String	None	No	No	NA

**Table 3–30 Data Source Input Values for Products Entity**

Data Source	Input Column	Type	Input Value
Products DS	ROW_ID	String	Row Id

---

**Note:** The Products Entity has caching enabled.

---

### 3.3.9 Promo Category Catalog Entity

**Table 3–31** *Promo Category Catalog Entity*

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Promotion Catalog Id	No	String	None	No	No	See Promotions Entity
Promotion Catalog Name	No	String	Catalog DS.NAME	No	No	Populated based on Id value of Promotion Catalog Id value
Promotion Category Name	No	String	None	No	No	See Promotions Entity

**Table 3–32** *Data Source Input Values for Promo Category Catalog Entity*

Data Source	Input Column	Type	Input Value
Catalog DS	ROW_ID	String	Promotion Catalog Id

### 3.3.10 Promotions Entity

**Table 3–33** *Promotions Entity*

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Catalog Array	Yes	String	None	No	No	NA
Catalog String	No	String	None	No	No	Populated through the Set Catalog String function which uses the Row Id as input
Category Array	Yes	String	None	No	No	NA
Category Catalog	Yes	Promo Category Catalog	None	No	No	NA
Category Catalog. Promotion Catalog Id	No	String	Catalog Category DS. CTLG_ID	No	No	Populated through the Catalog Lookup function
Category Catalog. Promotion Catalog Name	No	String	None	No	No	See Promo Category Catalog Entity for mapping
Category Catalog. Category Name	No	String	Catalog Category DS. NAME	No	No	Populated through the Category Lookup function
Category String	No	String	None	No	No	Populated through the Set Category String function which uses the Row Id as input
Description Text	No	String	None	No	No	See Promotions List Entity for mapping

**Table 3–33 (Cont.) Promotions Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Image Extension	No	String	Image DS.FILE_EXT	No	No	NA
Image File Name	No	String	Image DS.FILE_NAME	No	No	NA
Image Id	No	String	None	No	No	See Promotions List Entity for mapping
Image Name with Ext	No	String	None	No	No	Defaulted with the value returned from the GetImageName() function
Inclusive Eligibility Flag	No	String	Products DS. INCLSV_ELIG_RL_FLG	No	No	NA
Name	No	String	None	No	No	See Promotions List Entity for mapping
Row Id	No	String	None	No	No	See Promotions List Entity for mapping
Type	No	String	None	No	No	Default value is Promotion

**Table 3–34 Data Source Input Values for Promotions Entity**

Data Source	Input Column	Type	Input Value
Image DS	ROW_ID	String	Image Id
Products DS	ROW_ID	String	Row Id
Promo Category Catalog DS	ROW_ID	String	Row Id

### 3.3.11 Promotions List Entity

**Table 3–35 Promotions List Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Promotions	Yes	Promotions	None	No	No	Promotions are extracted from the S_PROD_INT table via two static input values, ORDERABLE_FLG = "Y" and PROMO_TYPE_CD = "Bundled Promotions"
Promotions. Catalog String	No	String	None	No	No	Populated through the Set Catalog String function which uses the Row Id as input
Promotions. Category String	No	String	None	No	No	Populated through the Set Category String function which uses the Row Id as input
Promotions. Description Text	No	String	Promotions DS. DESC_TEXT	No	No	NA



**Table 3–35 (Cont.) Promotions List Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Promotions. Image Extension	No	String	None	No	No	See Promotions Entity for mapping
Promotions. Image File Name	No	String	None	No	No	See Promotions Entity for mapping
Promotions. Image Id	No	String	Promotions DS. PROD_IMAGE_ID	No	No	NA
Promotions. Image Name With Ext	No	String	None	No	No	Defaulted with the value returned from the GetImageName() function
Promotions. Inclusive Eligibility Flag	No	String	None	No	No	NA
Promotions. Name	No	String	Promotions DS.NAME	No	No	NA
Promotions. Row Id	No	String	Promotions DS.ROW_ID	No	No	NA
Promotions. Type	No	String	None	No	No	Default value as Promotion
Promotions. Category Catalog	Yes	Promo Category Catalog Entity	None	No	No	See Promotions Entity for mapping

**Table 3–36 Data Source Input Values for Promotions List Entity**

Data Source	Input Column	Type	Input Value
Promotions DS	ORDERABLE_FLG	String	Y
Promotions DS	PROMO_TYPE_CD	String	Bundled Promotions

### 3.3.12 Raw Assets Entity

**Table 3–37 Raw Assets Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Create Date	No	Date	None	No	No	Mapped on the Assets Entity
Product Id	No	String	None	No	No	Mapped on the Assets Entity

**Table 3–37 (Cont.) Raw Assets Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Product Line	No	String	Product Line DS. NAME	No	No	Name value on the S_PROD_LN table is found based on the value for the Product Line Id attribute tied to the row id on the product line table.
Product Line Id	No	String	Products DS. PR_PROD_LN_ID	No	No	NA
Product Name	No	String	Mapped on the Assets Entity	No	No	Actual name value for the asset product is extracted through a Product Lookup function which returns the Name of the product associated to the product id. (The name value is on the table S_PROD_INT from the Products Entity.)

**Table 3–38 Data Source Input Values for Raw Assets Entity**

Data Source	Input Column	Type	Input Value
Product Line DS	ROW_ID	String	Product Line Id
Products DS	ROW_ID	String	Product Id

### 3.3.13 Raw Web Interactions Entity

**Table 3–39 Raw Web Interactions Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Catalogs Visited	No	String	None	No	No	Populated through the Web Interactions Entity
Categories Visited	No	String	None	No	No	Populated through the Web Interactions Entity
Create Date	No	Date	None	No	No	Populated through the Web Interactions Entity
Promotion Clicked	No	String	None	No	No	Populated through the Web Interactions Entity

### 3.3.14 Web Interaction Entity

**Table 3–40 Web Interaction Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Account Id	No	String	None	No	No	Populated through Integration points and used as a reference to the S_ORG_EXT Table (Customer entity)
Catalog Id	No	String	None	No	No	Populated through Integration points and used as a reference to the S_CTLG Table (Catalog entity)
Catalogs Visited	Yes	String	None	Yes	Yes	Derived attribute populated through the function BuildCategory() thread that is called from the Get Promotions Advisor. This attribute will store in array every catalog visited by the customer during the RTD session.
Categories Visited	Yes	String	None	Yes	Yes	Derived attribute populated through the function BuildCategory() thread that is called from the Get Promotions Advisor. This attribute will store in array every category visited by the customer during the RTD session.
Category Id	No	String	None	No	No	Populated through Integration points and used as a reference to the S_CTLG_CAT Table (Category entity)
Contact Id	No	String	None	No	No	Populated through Integration points and used as a reference to the S_CONTACT Table (Contact entity)
Current Catalog Name	No	String	Catalog DS.NAME	Yes	Yes	Lookup to the S_CTLG table through the Catalog Lookup function
Current Category Hierarchy	Yes	String	None	Yes	Yes	Populated through the Build Category Thread function
Current Category Name	No	String	Catalog Category DS. NAME	Yes	Yes	Lookup to the S_CTLG_CAT table through the Category Lookup function
Current Category Thread	No	String	None	Yes	Yes	Populated through the Build Category Thread function

**Table 3–40 (Cont.) Web Interaction Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Days Since Last Web Visit	No	Integer	None	Yes	Yes	Derived attribute populated through the ProcessWebInteractions() function. Identifies the days since the last web visit recorded by the RTD_WEB_HISTORY Table.
Hot Offers Page Hits	No	Integer	None	Yes	Yes	Derived attribute populated through the logic associated with the Presented Offers informant. This attribute records a count of how many times the Hot offers page has been seen.
Last Promotion Clicked	No	String	None	Yes	Yes	Derived attribute populated through the ProcessWebInteractions() function. Identifies the value of the promotion clicked on the last web page visit.
Last Visited Catalogs	Yes	String	None	Yes	Yes	Derived attribute populated through the ProcessWebInteractions() function. Identifies the catalogs visited on the last web visit.
Last Visited Categories	Yes	String	None	Yes	Yes	Derived attribute populated through the ProcessWebInteractions() function. Identifies the categories visited on the last web visit.
Locale	No	String	None	Yes	Yes	Populated through Integration points
Login Id	No	String	None	No	No	Populated through Integration points and used as a reference
Number of Visits in Past 30 Days	No	Integer	None	Yes	Yes	Derived attribute populated through the ProcessWebInteractions() function. Identifies the number of web visits for a given customer over the past 30 days.
Number of Visits in past 7 Days	No	Integer	None	Yes	Yes	Derived attribute populated through the ProcessWebInteractions() function. Identifies the number of web visits for a given customer over the past 7 days.
Promotion Clicked	No	String	None	No	No	Populated through the logic contained in the Offer Clicked informant
Site	No	String	None	Yes	Yes	NA

**Table 3–40 (Cont.) Web Interaction Entity**

Attribute	Array	Type	Mapping	Show In DC	Use for Analysis	Comments
Past Web Interactions	Yes	Raw Web Interactions	None	No	No	Array List of Web Interactions extracted from the Raw Assets Entity (see PastWeb attributes below for breakdown)
PastWeb. Catalogs Visited	No	String	RTD_WEB_HISTORY.CATALOGS_VISITED	No	No	Value used for processing in the ProcessWebInteractions() function
PastWeb. Categories Visited	No	String	RTD_WEB_HISTORY.CATEGORIES_VISITED	No	No	Value used for processing in the ProcessWebInteractions() function
PastWeb. Create Date	No	Date	RTD_WEB_HISTORY.CREATED	No	No	Value used for processing in the ProcessWebInteractions() function
PastWeb. Promotion Clicked	No	String	RTD_WEB_HISTORY.OFFER_CLICKED	No	No	Value used for processing in the ProcessWebInteractions() function

**Table 3–41 Data Source Input Values for Web Interaction Entity**

Data Source	Input Column	Type	Input Value
Web History DS	ACCOUNT_ID	String	Account Id

## 3.4 Informants

Informants are Integration Points used to pass information from an external application to Oracle RTD.

This section contains the following topics:

- [Section 3.4.1, "Initial Context"](#)
- [Section 3.4.2, "Offer Clicked"](#)
- [Section 3.4.3, "Presented Offers"](#)
- [Section 3.4.4, "Update Customer Profile"](#)
- [Section 3.4.5, "Close Session"](#)

### 3.4.1 Initial Context

The Initial Context Informant is used to start the Oracle RTD session. [Table 3–42](#) describes the parameters for the Informant Initial Context.

**Table 3–42 Informant Initial Context**

Parameter	Description
Informant Name	Initial Context
Session Keys	Session/Web Session ID

**Table 3–42 (Cont.) Informant Initial Context**

Parameter	Description
Request Data	Locale (String) - Mapped to Web Interaction Entity, Locale Attribute Site (String) - Mapped to Web Interaction Entity, Site Attribute
External System	Web Self Service
Order	0
Force session close	No
Logic	None
Pre-condition	None

### 3.4.2 Offer Clicked

The Offer Clicked Informant is to indicate which promotion was clicked through if any. [Table 3–43](#) describes the parameters for the Informant Offer Clicked.

**Table 3–43 Informant Offer Clicked**

Parameter	Description
Informant Name	Offer Clicked
Session Keys	Session/Web Session ID
Request Data	Choice Type (String) Promotion Id (String)
External System	Web Self Service
Order	0
Force session close	No
Logic	Logic is included to update the Promotion Acceptance Model for the promotion that was clicked through (positive outcome event)
Pre-condition	None

### 3.4.3 Presented Offers

The Presented Offers Informant is used to indicate which promotions are presented in the Web UI after Siebel E-Commerce pushes the ranked offers through the Eligibility engine. [Table 3–44](#) describes the parameters for the Informant Presented Offers.

**Table 3–44 Informant Presented Offers**

Parameter	Description
Informant Name	Presented Offers
Session Keys	Session/Web Session ID
Request Data	Choice Type (String) Promotion Id (String)
External System	Web Self Service
Order	0
Force session close	No

**Table 3–44 (Cont.) Informant Presented Offers**

Parameter	Description
Logic	Logic is included to update the Promotion Acceptance Model for the promotions that were presented (base event)
Pre-condition	None

### 3.4.4 Update Customer Profile

The Update Customer Profile Informant is used to send information related to the customer (Account/Contact) that explicitly logs in to the Web site. [Table 3–45](#) describes the parameters for the Informant Presented Offers.

**Table 3–45 Informant Update Customer Profile**

Parameter	Description
Informant Name	Update Customer Profile
Session Keys	Session/Web Session ID
Request Data	Account Id (String) - Mapped to Web Interaction Entity, Account Id Attribute Contact Id (String) - Mapped to Web Interaction Entity, Contact Id Attribute Login Id (String) - Mapped to Web Interaction Entity, Login Id Attribute
External System	Web Self Service
Order	0
Force session close	No
Logic	Logic is included to fill customer profile entities after the customer id is known
Pre-condition	None

### 3.4.5 Close Session

The Close Session Informant is used to end the Oracle RTD session with that specific web session id. [Table 3–46](#) describes the parameters for the Informant Close Session.

**Table 3–46 Informant Close Session**

Parameter	Description
Informant Name	Close Session
Session Keys	Session/Web Session ID
Request Data	None
External System	Web Self Service
Order	0
Force session close	Yes
Logic	None
Pre-condition	None

## 3.5 Advisors

Advisors are Integration Points by which an external application sends information to, and receives information back from, Oracle RTD.

This section contains the following topics:

- [Section 3.5.1, "Get Promotions"](#)

### 3.5.1 Get Promotions

The Get Promotions Advisor is used to send promotions (choices) to the web client. [Table 3–47](#) describes the parameters for the Advisor Get Promotions.

**Table 3–47** *Advisor Get Promotions*

Parameter	Description
Advisor Name	Get Promotions
Session Keys	Session/Web Session ID
Request Data	Catalog Id (String) - Mapped to Web Interaction Entity, Catalog Id Attribute Category Id (String) - Mapped to Web Interaction Entity, Category Id Attribute
Decision	Return Promotions (Both Control Group and Decision Group)
External System	Web Self Service
Order	0
Force session close	No
Logic	None
Pre-condition	Logic is included to build out the "category thread" for the given category input from the Advisor call. The Category thread outlines the category path the user would have had to go down in order to get to the category where the Advisor call was made.

## 3.6 Choice Groups and Choices

The RTD Siebel ECommerce Inline Service contains one choice group, Promotions, which has two sub-choice groups:

- Siebel Promotions, for dynamic choices
- RTD Promotions, for static choices

Each of these choice groups has the same choice attributes.

This section contains the following topics:

- [Section 3.6.1, "Promotions Choice Group"](#)
- [Section 3.6.2, "Siebel Promotions Choice Group \(Dynamic Choices\)"](#)
- [Section 3.6.3, "RTD Promotions Choice Group \(Static Choices\)"](#)

### 3.6.1 Promotions Choice Group

[Table 3–48](#) shows the component elements of the top-level choice group, Promotions.



**Table 3–48 Promotions Choice Group**

Tab	Parameter	Description
Choice Attributes	Image Name	String place holder used in the sub-choice groups Siebel Promotions and RTD Promotions.
Choice Attributes	Inclusive Eligibility Flag	String place holder used in the sub-choice groups Siebel Promotions and RTD Promotions.
Choice Attributes	Promotion Description	String place holder used in the sub-choice groups Siebel Promotions and RTD Promotions.
Choice Attributes	Promotion Id	String place holder used in the sub-choice groups Siebel Promotions and RTD Promotions.
Choice Attributes	Promotion Name	String place holder used in the sub-choice groups Siebel Promotions and RTD Promotions.
Choice Attributes	Promotion Type	String place holder used in the sub-choice groups Siebel Promotions and RTD Promotions.
Scores	Predicted by Likelihood of Acceptance for Promotions Acceptance Model	Utilizes the Performance Goal Maximize Acceptance Likelihood.
Choice Events	Clicked Through	NA
Choice Events	Presented	NA
Choice Eligibility	None	NA
Group Attributes	None	NA
Group Eligibility	None	NA
Dynamic Choices	None	NA

### 3.6.2 Siebel Promotions Choice Group (Dynamic Choices)

The Siebel Promotions choice group uses dynamic choices. The Siebel Promotions choice attributes are populated through the Promotions Entity, which in turn gets its values from the table S\_PROD\_INT.

[Table 3–49](#) shows the component elements of the Siebel Promotions choice group.

**Table 3–49 Siebel Promotions Choice Group (Dynamic Choices)**

Tab	Parameter	Description
Choice Attributes	Catalog String	Populated through the Promotions Entity, Catalog String attribute
Choice Attributes	Category String	Populated through the Promotions Entity, Category String attribute
Choice Attributes	Choice Type	Value=Dynamic

**Table 3–49 (Cont.) Siebel Promotions Choice Group (Dynamic Choices)**

Tab	Parameter	Description
Choice Attributes	Image Extension	Populated through the Promotions Entity, Image Extension attribute
Choice Attributes	Image Name	Populated through the Promotions Entity, Image Name with Ext attribute
Choice Attributes	Inclusive Eligibility Flag	Populated through the Promotions Entity, Inclusive Eligibility Flag attribute
Choice Attributes	Promotion Description	Populated through the Promotions Entity, Description Text attribute
Choice Attributes	Promotion Id	Populated through the Promotions Entity, Row Id attribute
Choice Attributes	Promotion Name	Populated through the Promotions Entity, Name attribute
Choice Attributes	Promotion Type	Populated through the Promotions Entity, Type attribute
Choice Attributes	Promotions Entity	Declaration of the Promotions entity used in order to map promotions entity attributes to their corresponding choice attribute
Scores	Predicted by Likelihood of Acceptance for Promotions Acceptance Model	Utilizes the Performance Goal Maximize Acceptance Likelihood
Choice Events	Clicked Through	NA
Choice Events	Presented	NA
Choice Eligibility	<i>(As released, both current catalog and current category rules are set. See Description.)</i>	To select one of the rules, follow the instructions in <a href="#">Section 4.1, "Utilizing the Category/Catalog Eligibility Rules."</a>
Group Attributes	Promotions Array Entity	NA
Group Eligibility	None	NA
Dynamic Choices	Use Dynamic Choices for this Choice Group	Yes
Dynamic Choices	Group attribute containing the List of entities for Choice	Promotions Array
Dynamic Choices	Choice attribute to assign the entity data	Promotions Entity
Dynamic Choices	Entity attribute that contains the choices id	Row Id
Dynamic Choices	Distribution mode for choices over choice group folders	Spill

**Table 3–49 (Cont.) Siebel Promotions Choice Group (Dynamic Choices)**

Tab	Parameter	Description
Dynamic Choices	Maximum number of choices within one choice group folder on decision center	100

### 3.6.3 RTD Promotions Choice Group (Static Choices)

Static choices are choices that are defined in Oracle RTD. For the static choice group RTD Promotions, the choice attributes have been set up, but no values are assigned to any static choice. This static choice group serves as a template for future customer use.

[Table 3–50](#) shows the component elements of the RTD Promotions choice group.

**Table 3–50 RTD Promotions Choice Group (Static Choices)**

Tab	Parameter	Description
Choice Attributes	Choice Type	Value=Static
Choice Attributes	Image Name	Not populated
Choice Attributes	Inclusive Eligibility Flag	Not populated
Choice Attributes	Promotion Description	Not populated
Choice Attributes	Promotion Id	Not populated
Choice Attributes	Promotion Name	Not populated
Choice Attributes	Promotion Type	Not populated
Scores	Predicted by Likelihood of Acceptance for Promotions Acceptance Model	Utilizes the Performance Goal Maximize Acceptance Likelihood
Choice Events	Clicked Through	NA
Choice Events	Presented	NA
Choice Eligibility	None	NA
Group Attributes	None	NA
Group Eligibility	None	NA
Dynamic Choices	None	NA

## 3.7 Decisions

There is one decision in the RTD Siebel ECommerce Inline Service, namely Return Promotions.

[Table 3–51](#) shows the component elements of the RTD Promotions choice group.

**Table 3–51** *Return Promotions Decision*

Parameter	Description
Select Choices From:	Siebel Promotions
Number of Choice to Select:	1000
Select at Random:	No
Segment Segments:	None
Priority for Default Segment:	Maximize Acceptance Likelihood: 100%
Pre-Decision Logic	None
Post Decision Logic	None

## 3.8 Models

There is one Choice Event model in the RTD Siebel ECommerce Inline Service, namely Promotions Acceptance Model. This models the "click through" likelihood of promotions presented through the Siebel E-Commerce site.

[Table 3–52](#) shows the component elements of the Promotions Acceptance Model.

**Table 3–52** *Promotions Acceptance Model*

Component/Parameter	Value
Use for prediction	Yes (Check)
Randomize Likelihood	Yes (Check)
Time Window	Month
Choice Group	Promotions
Base Event	Presented
Positive Outcome Events:	Clicked Through
Learn Location	End of Session

## 3.9 Functions

[Table 3–53](#) describes the functions in the RTD Siebel ECommerce Inline Service.

**Table 3–53 Functions to Support RTD Siebel ECommerce Inline Service**

Function	Inputs	Outputs	Area Utilized In	Comments
Build Category Thread	None	None	Used in the Get Promotions Advisor login	This function takes the category id that has been sent in from the informant and builds the "category hierarchy" for it based on what has been setup in the Siebel Categories view.
Catalog/Category Eligibility	Catalog/Category Name (String), Promotion String (String)	(Boolean)	Can be used an optional eligibility function in Choice Eligibility	This function compares either the catalog or category string tied to a promotion with a inputted catalog or category name and return TRUE if the case holds.
Catalog Id Lookup	Row Id (String)	Catalog Id (String)	Utilized within the Promotions Entity	This function extracts back from the cached Category Entity, the catalog Id associated with the inputted category id.
Catalog Lookup	Row Id (String)	Catalog Name (String)	Utilized within the Web Interaction Entity	This function extracts back from the cached Catalog Entity, the catalog name associated with the inputted catalog id.
Category Array Eligibility	Category Array (String Array), Promotion String (String)	(Boolean)	Can be used an optional eligibility function in Choice Eligibility	This function compares a given category string tied to a promotion with a inputted category array and return TRUE if one of the array values holds.
Category Lookup	Row Id (String)	Category Name (String)	Utilized in the Promotions and Web Interaction Entity as well as the BuildCategoryThread() function.	This function extracts back from the cached Category Entity, the category Name associated with the inputted category id.
Days Since	InputDate (Date)	DaysSince (Integer)	Utilized in the ProcessAssets() and ProcessWebInteractions() functions	This function calculates the delta between the current system date with any given input date.
Fill Entities	None	None	Used in the Update Customer Profile Informant logic section	The Fill Entities function is used as a container function which calls all of the logic/additional functions used to populate the session entities for this Inline Service.
Get Image Name	File Name (String), File Ext (String)	File Name (String)	Used in the Promotions Entity	This function concatenates the file name and extension of if a given image record.
Get Siebel Promotions	Type (String)	Promotions (Promotions Entity)	Using the Siebel Promotions Choice Group (Group attributes tab)	This function populated the Promotions List Entity with an array of Promotions extracted back from Siebel.
Industry Lookup	Row Id (String)	Industry Name (String)	Used in the Customer Entity	This function returns back the name of an industry for a given industry id.

**Table 3–53 (Cont.) Functions to Support RTD Siebel ECommerce Inline Service**

Function	Inputs	Outputs	Area Utilized In	Comments
Insert Message Result	Session Id (String), Account Id (String), Offer Clicked (String), Categories Visited (String), Catalogs Visited (String)	None	Used in the Close Session informant logic	This function inserts rows into the RTD_WEB_HISTORY table - see Log Writer function.
Log Write	None	TableInserter	Used as the default value for the Application attribute, Log Writer)	This function instantiates a tableInserter and returns handle for the table RTD_WEB_HISTORY.
Multiply	A (double), B (Double)	A * B (Double)	Optional function	This function is used to multiply two inputted values.
Number Is Null	Number (Integer)	(Boolean)	Optional function used to check if any integer attributes are null	This function returns True if number value provided is Null, otherwise False.
Parent Category Lookup	Row Id (String)	Parent Category Id (String)	Used in the BuildCategoryThread() function	This function extracts back from the cached Category Entity, the Parent Category Id associated with the inputted category id. This is in turn used in the Build category thread function.
Process Assets	None	None	Utilized in the Fill Entities() function	This function loops through the asset records associated with an account to determine and set which products/product lines an Account owns.
Process Web Interactions	None	None	Utilized in the Fill Entities() function	This function processes the web interaction records Source tied to a given account id and populates the appropriate Web Interaction entity attributes.
Product Lookup	Row Id (String)	Name (String)	Utilized in the Assets Entity	This function returns the product name for a given product record.
Return SDOId	Choice (choice)	(String)	Optional function which can be used for choices	This function is used to return the SDOId (Internal Name) of the choice inputted in.
Return System Date	None	Current Time (Date)	Used in the Session Entity	This function is used to return the current system date/time.
Set Catalog String	Promo Id (String)	Catalog String (String)	Used in the Promotions Entity	This function populates the Promotions Entity attribute, Catalog String based on the category records that are associated to the promotion in Siebel.

**Table 3–53 (Cont.) Functions to Support RTD Siebel ECommerce Inline Service**

Function	Inputs	Outputs	Area Utilized In	Comments
Set Category String	Promo Id (String)	Category String (String)	Used in the Promotions Entity	This function populates the Promotions Entity attribute, Category String based on the category records that are associated to the promotion in Siebel.
To Integer	String (String)	(integer)	Optional Transformation function	This function converts a string value into an integer.
To String	Integer (integer)	(String)	Optional Transformation function	This functions converts an integer value into a string.





---

## Configuring the RTD Siebel ECommerce Inline Service

This chapter outlines the steps that you can take to configure the RTD Siebel ECommerce Inline Service.

These steps are not an exhaustive list of the modifications that users can perform in the RTD Siebel ECommerce Inline Service. However, they represent the most common tasks for users to perform to align the Inline Service to their specific business process requirements.

This chapter contains the following topics:

- [Section 4.1, "Utilizing the Category/Catalog Eligibility Rules"](#)
- [Section 4.2, "Modifying Promotion Attributes Sent to Client"](#)
- [Section 4.3, "Adding More Oracle RTD Data Sources"](#)
- [Section 4.4, "Adding Attributes to Existing Entities"](#)
- [Section 4.5, "Modifying Scoring Logic"](#)
- [Section 4.6, "Enabling the RTD\\_WEB\\_HISTORY Table"](#)

### 4.1 Utilizing the Category/Catalog Eligibility Rules

As part of the provided Inline Service, Oracle RTD provides an eligibility function, Catalog/Category Eligibility. This eligibility function enables Oracle RTD to determine which promotions from Siebel are eligible, based on either the current category or catalog that the user is on within the E-Commerce Web site. This function is part of the RTD Siebel ECommerce Inline Service metadata and is defined in the Choice Eligibility tab of the Siebel Promotions choice group.

As released, the use of the RTD Catalog/Category Eligibility function is enabled for both the current catalog and current category that the user is on within the E-Commerce Web site. To disable the use of either of these functions, change the value of the **isEnabled** parameter to this function to False.

1. In Oracle RTD Studio, navigate to the Siebel Promotions choice group - which is located under the Promotions parent choice group.
2. Within Siebel Promotions, navigate to the Choice Eligibility tab.

The Choice Eligibility tab is where the Catalog/Category Eligibility rules are defined.

3. For the desired rule, set the value of the **isEnabled** parameter to False, if the rule is not be used.

4. You can identify the rule for *Catalog eligibility* by the rule input parameter, session/web interaction/current catalog name.

The corresponding identifier for the *Category eligibility* rule is the rule input parameter, session/web interaction/current category name.

After you have made your desired changes, you are ready to redeploy the Inline Service with the updated eligibility rules.

## 4.2 Modifying Promotion Attributes Sent to Client

The choice group Siebel Promotions, as released, has the following attributes that are sent to the client when the Get Promotions Advisor is called:

1. Image Name
2. Inclusive Eligibility Flag
3. Promotion Description
4. Promotion Id
5. Promotion Name
6. Promotion Type

To add a new attribute to be sent with the promotion, perform the following steps:

1. If necessary, add to the Promotions data source any corresponding data columns that store the desired promotion attributes.
2. For each new data source attribute, create a new attribute in the Promotions entity.
3. From the Promotions List Entity, if the new attribute is to be extracted from the Promotions data source, map the newly created attribute to the corresponding column.
4. On the Promotions choice group, create a new choice attribute corresponding to the entity attribute.
5. Map the entity attribute to the choice attribute, from the attribute Value column
6. Right click the entity attribute and select Properties.

From the Properties window, check the **Send to client** option.

---

**Note:** Additional configuration or customization may be required within the Siebel E-Commerce front end application to process or present any newly created attributes tied to an Oracle RTD promotion.

---

To remove a Promotion attribute from the set of those sent to the client, simply delete the appropriate choice attribute from the Promotions Choice group.

## 4.3 Adding More Oracle RTD Data Sources

Section 3.2, "Data Sources" describes which data sources have been defined in the released version of the RTD Siebel ECommerce Inline Service.

To add an additional data source, refer to the Oracle RTD platform documentation for steps that apply to your database.

## 4.4 Adding Attributes to Existing Entities

Section 3.3, "Entities" describes which data attributes have been defined in the released version of the RTD Siebel ECommerce Inline Service. These attributes are a combination of derived attributes and attributes mapped to values extracted directly from Siebel OLTP table columns.

In order to map additional columns directly from Siebel, perform the relevant steps as described in the following sections.

### For Data Sources already defined in the Inline Service

For each column that you want to add, perform the following steps:

1. Add the additional column as an output column to the appropriate data source.
2. In the Definition tab of the appropriate entity tied to that data source, create a new corresponding entity attribute.
3. From the Mapping tab of the appropriate entity, map the entity attribute to the desired data source column added in Step 1.

### For Mapping Columns from New Tables

1. Add the additional table as a new data source in the RTD Siebel ECommerce Inline Service.

Specify the output columns and input columns as appropriate. Refer to Oracle RTD platform documentation for instructions on how to add a data source.

2. If needed, create a new Oracle RTD entity.

Associate the new entity to either the Session Entity directly or to another Oracle RTD entity that is already part of the session.

3. From the Definition tab of either the new entity or the existing one, create the new, desired entity attributes.
4. From the Mapping tab of the appropriate entity, map each of the new entity attributes to the appropriate Data Source columns.

If necessary, specify the source value for the required input columns for that appropriate data source.

---

**Note:** If the desired table does not reside in the Siebel OLTP, refer to the Oracle RTD platform documentation for how to add a new JDBC data source first, before adding the table to the Inline Service.

---

### For Derived Attributes

1. Create the desired entity attribute within the appropriate entity.
2. You can either map the attribute directly to a data source column, or use a function to populate the attribute.

The function may be a transformation function or a new custom function that you create that assigns the value of this attribute.

You can add the function as the default value to the attribute, or you can create code which populates the attribute directly.

## 4.5 Modifying Scoring Logic

Scoring in the released version of the RTD Siebel ECommerce Inline Service is based on two performance goals, Maximize Acceptance Likelihood and Maximize Customer Consistency.

For the performance goal, Maximize Acceptance Likelihood, the Inline Service is set up to extract the likelihood value based on the calculated scores determined from the Promotions Acceptance Model.

For Maximize Customer Consistency, the scoring method has been left blank. This is to enable customers to add scoring rules to rank promotions according to their own initiatives (for example, hard coded rankings or outside scores) as opposed to the Oracle RTD models.

## 4.6 Enabling the RTD\_WEB\_HISTORY Table

The RTD\_WEB\_HISTORY table is a table in the Oracle RTD database created specifically for the Siebel E-Commerce solution. When enabled, this table stores the web transaction record of the Oracle RTD session when the Oracle RTD session is closed. Logic to write to this table appears in the "cleanup logic" section of the Session entity.

In the Inline Service, the Application parameter **Enable RTD Web History** controls whether the "cleanup logic" is executed.

To enable Oracle RTD to write to the RTD\_WEB\_HISTORY table, set the Application parameter **Enable RTD Web History** to True.

---

**Note:** In the released version of the RTD Siebel ECommerce Inline Service, the value of **Enable RTD Web History** is set to True.

---

### Creating the RTD\_WEB\_HISTORY Table

If the RTD\_WEB\_HISTORY table was not created as part of your installation, you will need to create the RTD\_WEB\_HISTORY table within the RTD database (SDDS or SDDb.)

For more information on how create the RTD\_WEB\_HISTORY table, refer to [Section 1.7](#) in the Installation chapter of this document.