Oracle® Real-Time Decisions Base Application

Installation and Reference Guide

Release 3.2

E52397-01

December 2014

Oracle Real-Time Decisions Base Application Installation and Reference Guide, Release 3.2

E52397-01

Primary Author: Oracle Corporation

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

Contents

Part I Getting Started

1 Overview of Base Application Modules

2 Installation of Oracle RTD Base Application Modules

3 Overview of Oracle Real-Time Decisions

Part II Base Inline Services

4 Base E-Commerce

5 Base Customer Service

6 Configuring the Base Inline Services

Part III Reference Library

7 Numeric Predictions

8 Batch Processing

Part IV Oracle RTD Decision Management Application Metadata and Inline Services

9 Elements in Core and Base Marketing Inline Services

10 Decision Management Setup Files

Preface

This document describes the features of the Oracle Real-Time Decisions (Oracle RTD) Base 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

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Real-Time Decisions platform version 3.0 documentation set and the Oracle Real-Time Decisions Base Application Release 3.1.1 documentation set:

- *Oracle Real-Time Decisions Installation and Administration Guide*
- *Oracle Real-Time Decisions Decision Studio Reference Guide*
- *Oracle Real-Time Decisions Decision Center User's Guide*
- *Oracle Real-Time Decisions Release Notes*
- *Oracle Real-Time Decisions Base Application Decision Management Installation and Configuration Guide*
- *Oracle Real-Time Decisions Base Application Decision Management Applications User's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.

Convention	Meaning
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Getting Started

Part 1 helps you get started with Oracle RTD Base Application, and contains the following chapters:

- Chapter 1, "Overview of Base Application Modules"
- Chapter 2, "Installation of Oracle RTD Base Application Modules"
- Chapter 3, "Overview of Oracle Real-Time Decisions"

Overview of Base Application Modules

The Oracle RTD Base Application consists of two types of application module, that together make up Oracle RTD best practices to apply on the Oracle RTD platform:

- Base Inline Services, which are complete Inline Services with end to end workflows
- Reference Library, whose modules each demonstrate a specific piece of Oracle RTD functionality

Base Inline Services

Base Inline Services are complete Inline Services that each address a general workflow. They can be customized to support an integration of Oracle RTD into a customer's own specific workflow, for example, E-Commerce Workflow, Contact Center Workflow, and so on.

Unlike specialized solutions, such as Siebel Intelligent Offer Generation with Oracle RTD, the Base Inline Services do not rely on any specific front end application or data schema. Instead, it is up to customers to configure the components of the Inline Services to align with their own front end user interfaces and customer data sources. This allows customers the full flexibility of applying the Oracle RTD best practices to their unique applications.

With the Base Inline Services, multiple use cases for Oracle RTD are pre-configured for customers to choose from, for decisioning and prediction as well as for real time analysis. Ultimately, customers may choose to use only a subset of a Base Inline Service to begin their Oracle RTD implementation, and utilize additional components in a later phase.

The Base Inline Service modules available with Oracle RTD Base Application are the following:

- Base E-Commerce

- Base Customer Service

For more information, see Part II, "Base Inline Services".

Reference Library Inline Services

Reference Library Inline Services are fully implementable Inline Services that demonstrate how to enable a desired piece of functionality within Oracle RTD.

Unlike the Base Inline Services that are in the Base Application, these examples do not support an end to end customer workflow and but rather demonstrate only one specific piece of functionality. However, these examples are aligned with components of the Base Inline Services to draw analogies as to how the components can be applied in an implemented solution.

Examples of Reference Library functionality supported by the Oracle RTD Base Application include numeric predictions in Oracle RTD and Oracle RTD batch implementation.

The Reference Library modules available with Oracle RTD Base Application are the following:

- Numeric Predictions
- Batch Processing

For more information, see Part III, "Reference Library"

Installation of Oracle RTD Base Application Modules

This chapter describes how to install an Oracle RTD Base Application module.

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 Guide*, which is available with the Oracle RTD platform software.

2.1 Installation Steps

Setting up an Oracle RTD Base Application module involves importing the Inline Service for the module into Oracle RTD Studio, and then configuring it to align with the customer's data sources and business applications.

To set up the module Inline Service, you must import the module Inline Service project into Oracle RTD 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. Locate the zip file for the Base Application module that you want to install.

   The zip file is located in one of the following directories:

   - `\software\Oracle Real-Time Decisions Base Application\Base Inline Services\`
     `<module_directory_name>`

- `\software\Oracle Real-Time Decisions Base Application\Reference Library\`
  `<module_directory_name>`

  depending on whether the Inline Service for the module is a Base Inline Service or a Reference Library Inline Service. For more details, see Chapter 1, "Overview of Base Application Modules."

3. Unzip the Base Application module zip file, for example `RTD_Base_ECommerce.zip`, Into *RTD_ILS_HOME*.

   Unzipping the file creates a directory under *RTD_ILS_HOME*.

   This sub-directory is referred to from now on as *MODULE_ILS_FOLDER*.

4. In Oracle RTD Studio, select File > Import.
5. From the Import window, select Existing Projects into Workspace.
6. If not already visible in the Projects area of the Import Projects dialog box, click the Browse button, and locate *RTD_ILS_HOME*\MODULE_ILS_FOLDER. For example, `C:\RTD_ILS\RTD_Base_ECommerce`.
7. Select the directory *MODULE_ILS_FOLDER* in the file dialog box and click Ok.
8. In the Projects area of the Import Projects dialog box, select the project name for the Base Application module.
9. Click the Finish button to load the project into Oracle RTD Studio.

The Base Application module project appears in the Inline Service Explorer view of Oracle RTD Studio.

Depending on how you want to use the modules, you may decide to configure the associated Inline Service or Inline Services according to your business process requirements. For more information, see Chapter 6, "Configuring the Base Inline Services."

3

Overview of Oracle Real-Time Decisions

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:

3.1 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
Minimize 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:

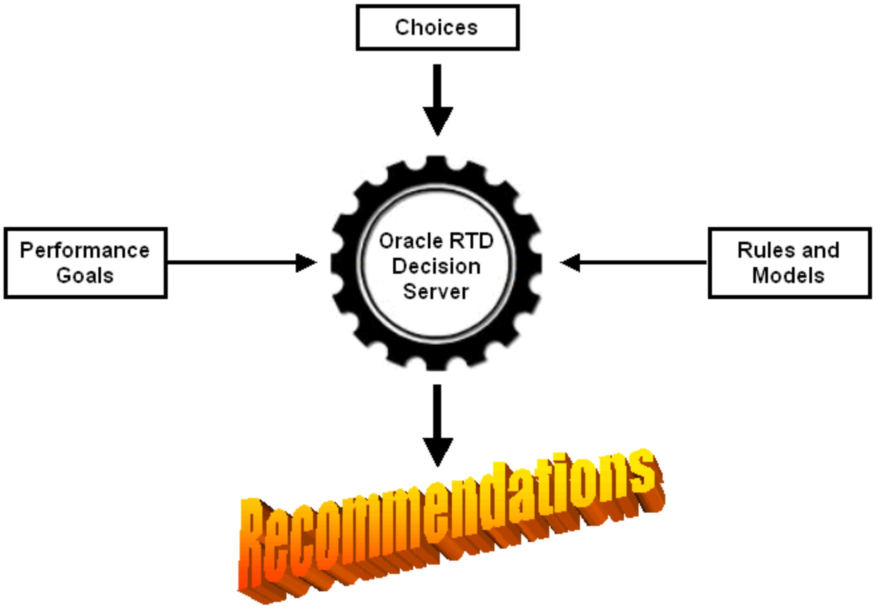
Company	Job	Location	Salary	Promotion Prospects	Choice	Score
VeriLeaf	Quality Manager	Green Acres	220,000	Good	**Accept VeriLeaf job**	**60**
PlentiSol	Research Director	Balmington	250,000	Fair	**Accept PlentiSol job**	**50**
FaunaFlex	Project Manager	North Elk	200,000	High	**Accept FaunaFlex job**	**80**

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 3.4, "The Oracle RTD Decision Process"

- Section 3.5, "More About the Oracle RTD Decision Process Elements"

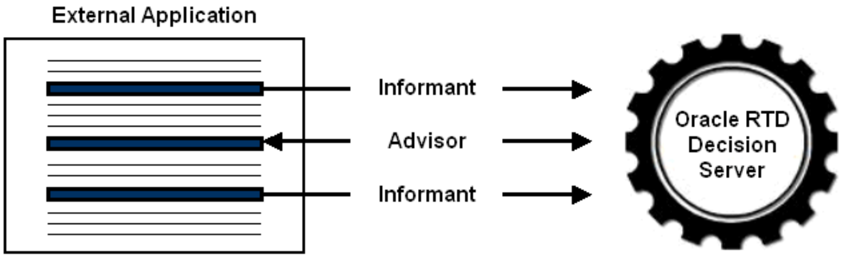
In general, Oracle RTD connects with other applications, and passes its recommendations to these external applications. See the next section for more information about how Oracle RTD integrates with external applications.

3.2 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.

3.3 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 3–1 shows, in outline form, how a typical application interfaces with Oracle RTD.

Figure 3–1 Overview of External Application Integration with Oracle RTD

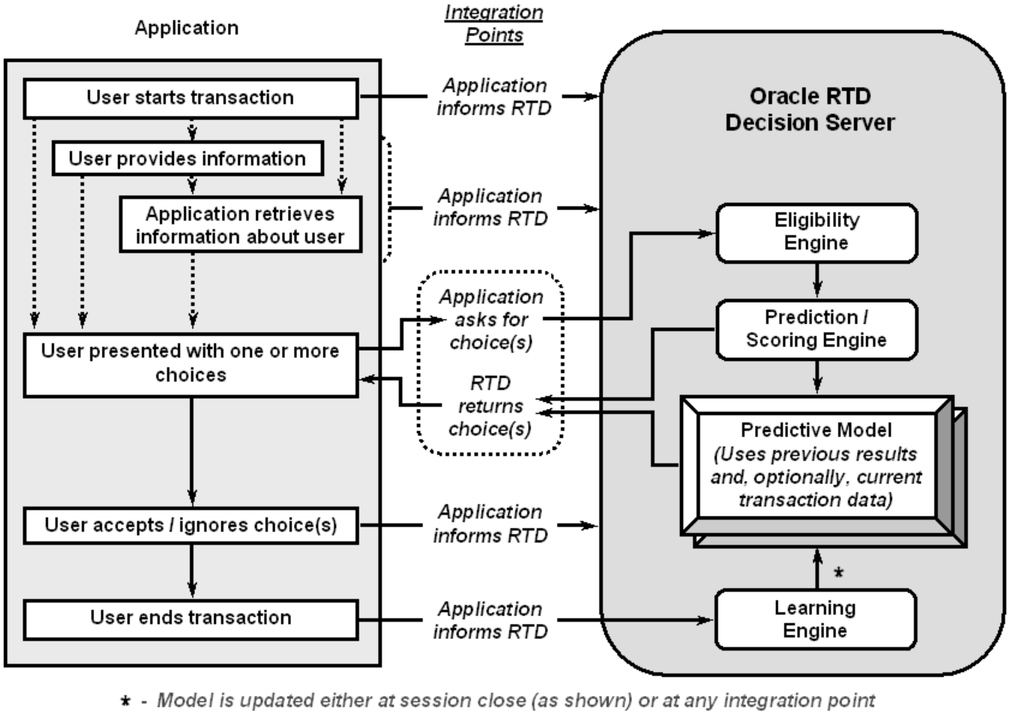

Application
Integration Points
User starts transaction
User provides information
Application retrieves information about user
User presented with one or more choices
User accepts / ignores choice(s)
User ends transaction
Application informs RTD
Application informs RTD
Application asks for choice(s)
RTD returns choice(s)
Application informs RTD
Application informs RTD
Oracle RTD Decision Server
Eligibility Engine
Prediction / Scoring Engine
Predictive Model (Uses previous results and, optionally, current transaction data)
Learning Engine
*
* - Model is updated either at session close (as shown) or at any integration point

Figure 3–1 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.

3.4 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 3.5, "More About the Oracle RTD Decision Process Elements."

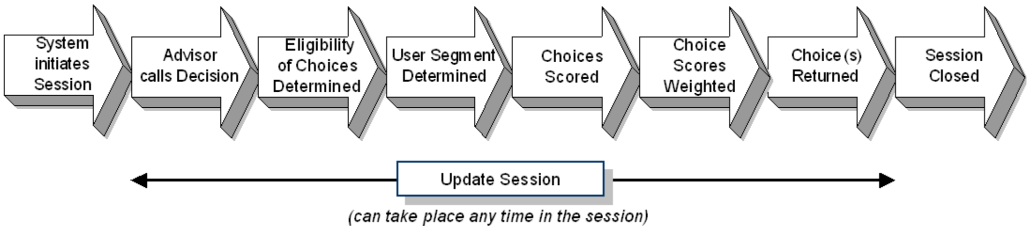
The Oracle RTD decision process is based on a framework that takes into account the following factors:

- 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 3–2 shows the basic Oracle RTD processes, which include session start and finish, as well as the Oracle RTD decision process steps.

Figure 3–2 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.

3.5 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

3.5.1 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 with 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.

3.5.2 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

3.5.3 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 the topic Dynamic Choices in the Externalized Objects Management chapter in *Oracle Real-Time Decisions Platform Developer's Guide*.

3.5.4 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.

3.5.5 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 with the Decision.

3.5.6 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.

3.5.7 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.

3.5.7.1 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 3.5.7.3, "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

3.5.7.2 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.

3.5.7.3 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.

3.6 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

3.6.1 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.

3.6.2 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 runtime 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.

3.6.3 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.

3.7 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.

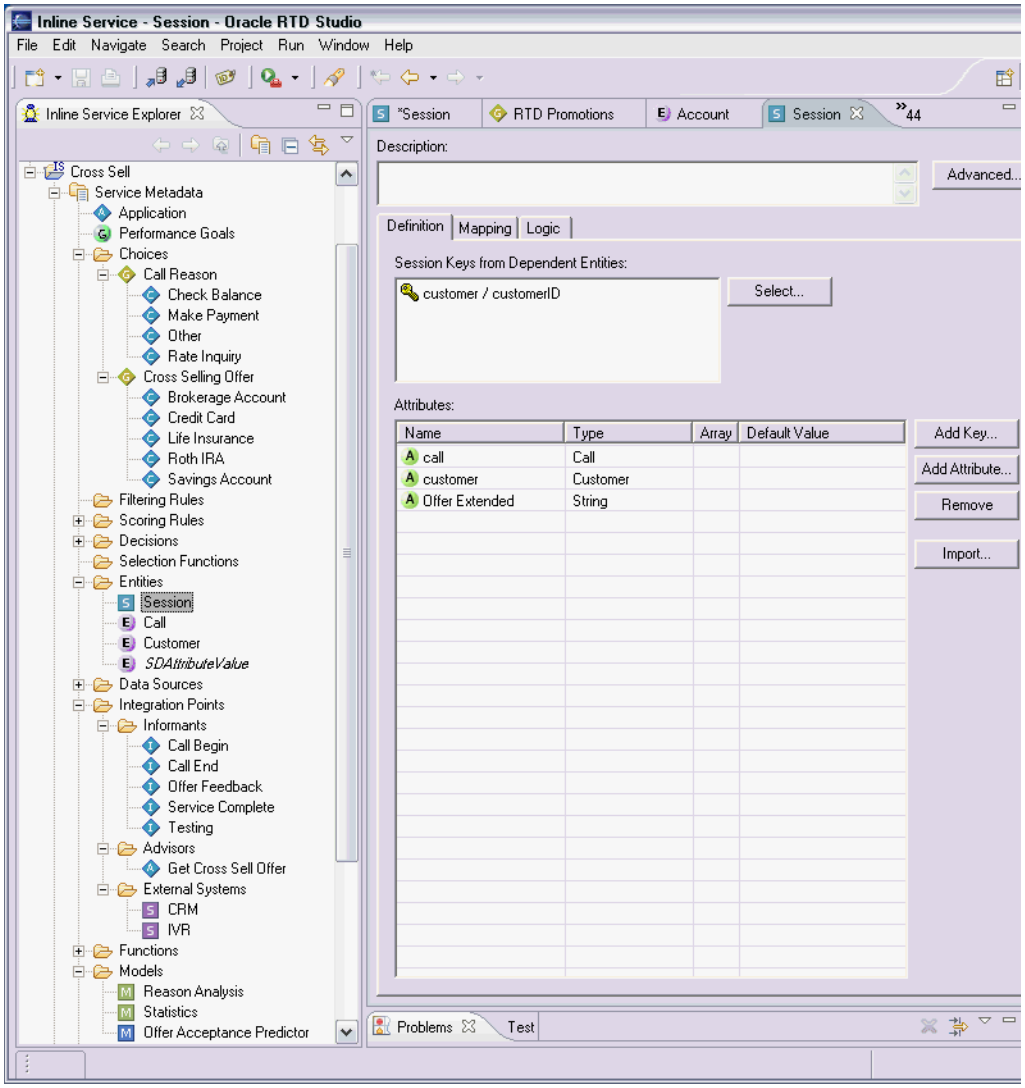
For more information on how to configure an Inline Service, see the Related Documents in the Preface chapter, in particular the *Oracle Real-Time Decisions Decision Studio Reference Guide*.

3.8 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 3–3 shows some of the elements of an Inline Service, called Cross Sell, as displayed in the Decision Studio.

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



customer / customerID

Name	Type	Array	Default Value
call	Call		
customer	Customer		
Offer Extended	String		

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

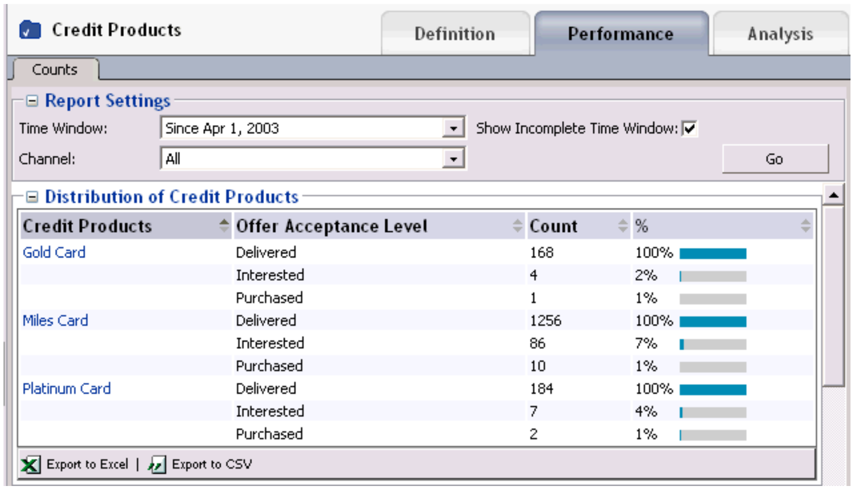
3.9 Analytic Reports and the Oracle RTD Decision Center

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**

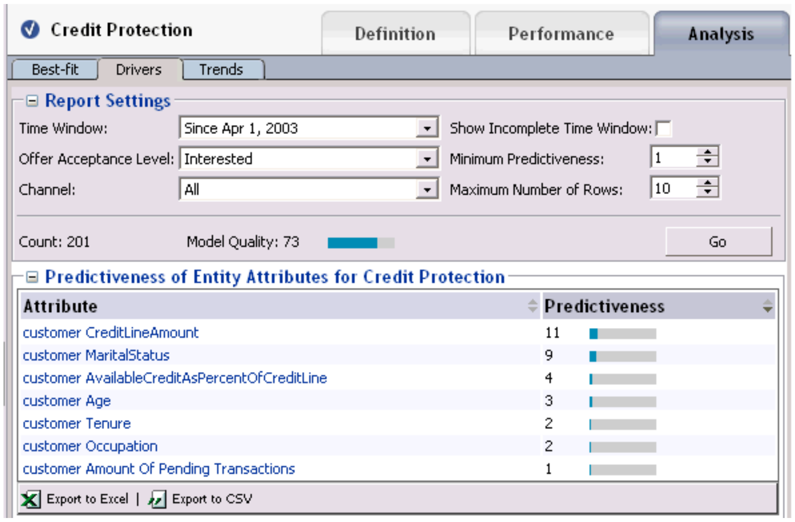


Time Window: Since Apr 1, 2003
Show Incomplete Time Window: ☑
Channel: All

Credit Products	Offer Acceptance Level	Count	%
Gold Card	Delivered	168	100%
	Interested	4	2%
	Purchased	1	1%
Miles Card	Delivered	1256	100%
	Interested	86	7%
	Purchased	10	1%
Platinum Card	Delivered	184	100%
	Interested	7	4%
	Purchased	2	1%

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

- **Choice Analysis Drivers**


Credit Protection
Definition
Performance
Analysis
Best-fit
Drivers
Trends
Report Settings
Go
Predictiveness of Entity Attributes for Credit Protection
Export to Excel | Export to CSV

Time Window: Since Apr 1, 2003
Show Incomplete Time Window: ☐
Offer Acceptance Level: Interested
Minimum Predictiveness: 1
Channel: All
Maximum Number of Rows: 10
Count: 201
Model Quality: 73

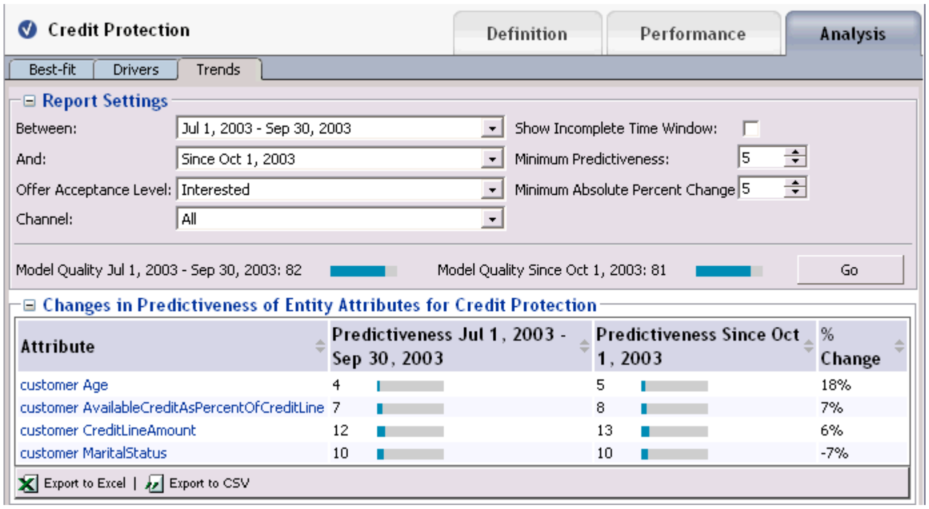
Attribute	Predictiveness
customer CreditLineAmount	11
customer MaritalStatus	9
customer AvailableCreditAsPercentOfCreditLine	4
customer Age	3
customer Tenure	2
customer Occupation	2
customer Amount Of Pending Transactions	1

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**



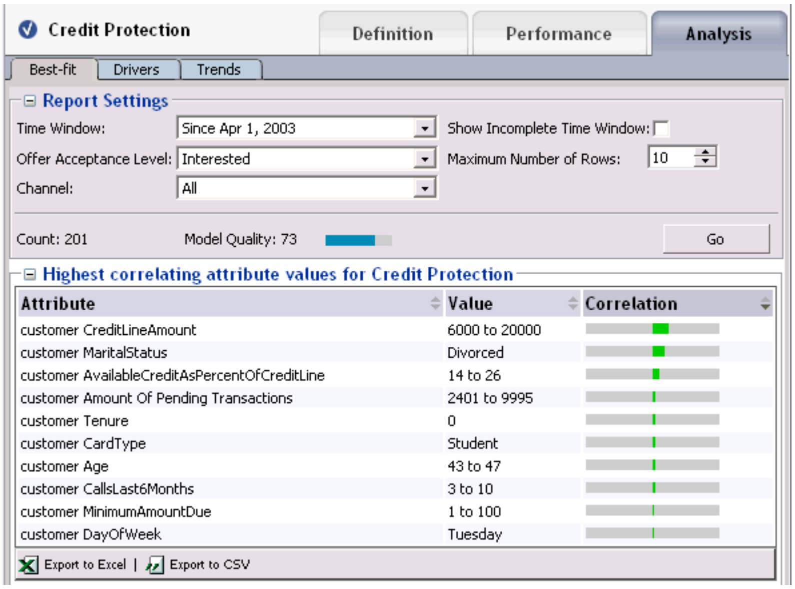
Between: Jul 1, 2003 - Sep 30, 2003
And: Since Oct 1, 2003
Offer Acceptance Level: Interested
Channel: All
Show Incomplete Time Window: ☐
Minimum Predictiveness: 5
Minimum Absolute Percent Change 5

Model Quality Jul 1, 2003 - Sep 30, 2003: 82
Model Quality Since Oct 1, 2003: 81

Attribute	Predictiveness Jul 1, 2003 - Sep 30, 2003	Predictiveness Since Oct 1, 2003	% Change
customer Age	4	5	18%
customer AvailableCreditAsPercentOfCreditLine	7	8	7%
customer CreditLineAmount	12	13	6%
customer MaritalStatus	10	10	-7%

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**



Time Window: Since Apr 1, 2003
Offer Acceptance Level: Interested
Channel: All
Show Incomplete Time Window: ☐
Maximum Number of Rows: 10

Count: 201
Model Quality: 73

Attribute	Value	Correlation
customer CreditLineAmount	6000 to 20000	
customer MaritalStatus	Divorced	
customer AvailableCreditAsPercentOfCreditLine	14 to 26	
customer Amount Of Pending Transactions	2401 to 9995	
customer Tenure	0	
customer CardType	Student	
customer Age	43 to 47	
customer CallsLast6Months	3 to 10	
customer MinimumAmountDue	1 to 100	
customer DayOfWeek	Tuesday	

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 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's Guide*.

3.10 Oracle RTD Batch Framework

Oracle RTD Batch Framework is a set of components that can be used to provide batch facilities in an Inline Service. This enables the Inline Service to be used not just for processing interactive Integration Point requests, but also for running a batch of operations of any kind. Typically, a batch will read a set of input rows from a database table, flat file, or spreadsheet, process each input row in turn, and optionally write one or more rows to an output table for each input row.

For general information on how the Batch Framework is used with Inline Services, see the Oracle RTD Batch Framework chapter in *Oracle Real-Time Decisions Platform Developer's Guide*.

The Oracle RTD Base Application includes two examples of the use of the batch framework in Oracle RTD Inline Services:

- The Base Application module **Base Customer Service** includes an example of batch operations that can be applied in general customer service applications.

  For more details, see Section 5.10, "Batch Use of the Base Customer Service Module."

- The Base Application module **Batch Processing** demonstrates the use of the batch framework as applied to the Cross Sell example Inline Service that is released with the Oracle RTD platform.

  For more details, see Chapter 8, "Batch Processing."

Base Inline Services

Part 2 describes the component elements of the Base Inline Services and provides general guidelines for how to configure the Base Inline Services to fulfill an organization's business requirements.

Part 2 contains the following chapters:

- Chapter 4, "Base E-Commerce"
- Chapter 5, "Base Customer Service"
- Chapter 6, "Configuring the Base Inline Services"

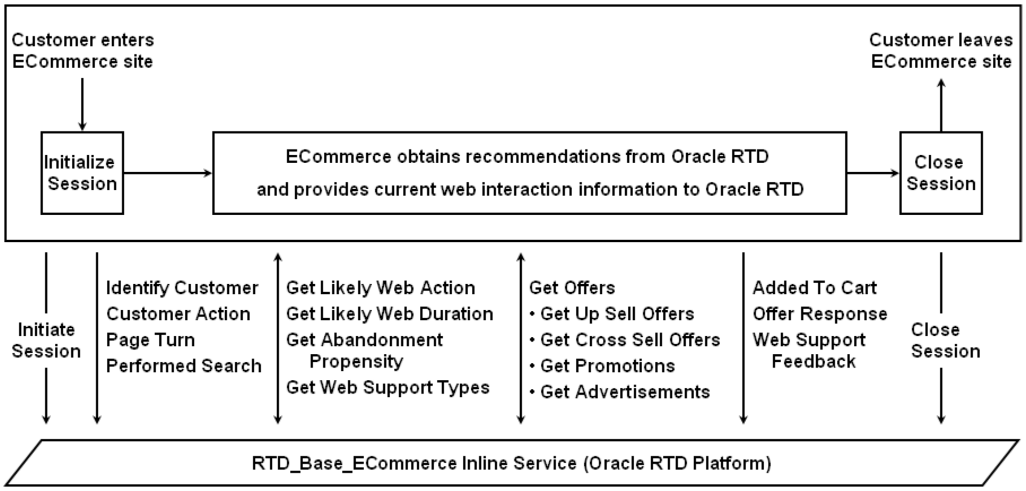
Base E-Commerce

This chapter describes the elements in the Inline Service RTD_Base_ECommerce. It contains the following topics:

- Section 4.1, "Introduction to Base E-Commerce"
- Section 4.2, "Overview of Choices Groups and Choices"
- Section 4.3, "Entities"
- Section 4.4, "Informants"
- Section 4.5, "Advisors and Associated Decisions, Choice Groups, and Models"
- Section 4.6, "Functions"

4.1 Introduction to Base E-Commerce

The following diagram shows an overview of the stages of a typical ECommerce application and its possible interactions with the Inline Service RTD_Base_ECommerce.



The RTD_Base_ECommerce Inline Service serves as a general ECommerce framework for customers to adapt to their business processes.

The RTD_Base_ECommerce Inline Service provides pre-defined entities, choices, decisions, models, and integration points.

From an end user perspective, the RTD_Base_ECommerce Inline Service is designed on the assumption that customers will perform additional configuration and further customization to fulfill their business requirements.

Additional tasks involve mapping the logical entity attributes to a customer's physical data sources and to develop the front-end environment for the presentation of any Oracle RTD recommendations.

For more information, see Chapter 6, "Configuring the Base Inline Services."

4.2 Overview of Choices Groups and Choices

This section presents an overview of the choice groups and choices, and their usage in the integration points.

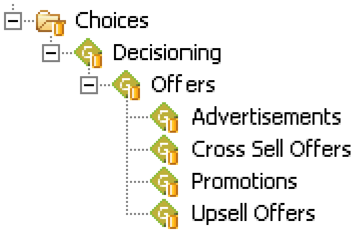
This section contains the following topics:

- Section 4.2.1, "Offer-Oriented Choices"
- Section 4.2.2, "Interaction-Oriented Choices"
- Section 4.2.3, "Informants Usage with Offer-Oriented and Transaction-Oriented Choices"

For more information about how the choices and choice groups are used, see Section 4.5, "Advisors and Associated Decisions, Choice Groups, and Models."

4.2.1 Offer-Oriented Choices

Choices for offer-oriented use cases are structured as follows:



Offers and all other choice groups under it are comprised of dynamic choices. The choice data is either supplied as an advisor's incoming parameter value (see Section 4.5, "Advisors and Associated Decisions, Choice Groups, and Models") or retrieved by Oracle RTD from external data sources.

Advisors for Offer-Oriented Choices

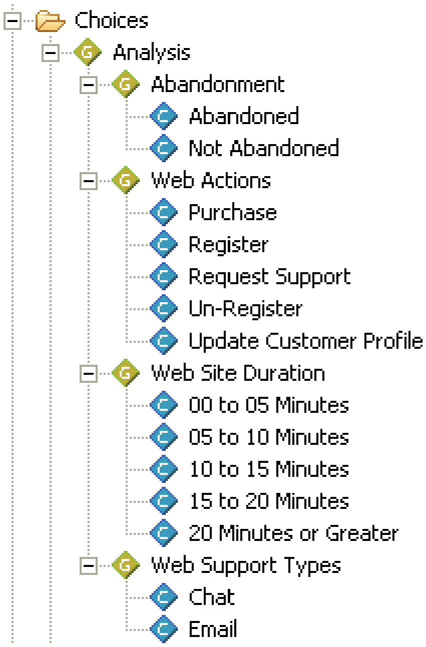
The offer-oriented choices are returned when invoking the following advisors:

- **Get Cross Sell Offers**
- **Get Upsell Offers**
- **Get Promotions**
- **Get Advertisements**
- **Get Offers**
  - Updates current web interaction attributes and Web Pages for the session.
  - Identifies any given page a customer navigates to that needs to be tracked.

For more specific information about the advisors, see Section 4.5, "Advisors and Associated Decisions, Choice Groups, and Models."

4.2.2 Interaction-Oriented Choices

Choice groups and choices for interaction-oriented usages are structured as follows. Note that the choices under each choice group can be reconfigured to suit the end users needs and serve as a template.



- Choices
  - Analysis
    - Abandonment
      - Abandoned
      - Not Abandoned
    - Web Actions
      - Purchase
      - Register
      - Request Support
      - Un-Register
      - Update Customer Profile
    - Web Site Duration
      - 00 to 05 Minutes
      - 05 to 10 Minutes
      - 10 to 15 Minutes
      - 15 to 20 Minutes
      - 20 Minutes or Greater
    - Web Support Types
      - Chat
      - Email

- **Abandonment**
  - Identifies if customers have abandoned their web session. This would be determined by the business based on their definitions of abandonment.

    Abandonment choice can be obtained by invoking the advisor **Get Abandonment Propensity**, and its prediction model is updated by invoking the advisor **Close Session**.
  - Returns likelihood of abandonment for a customer.
- **Web Actions**
  - Identifies key actions done by a customer that can later be applied to entity attributes or models, for example, Update Customer Profile, Register, Un-register. Chat request.

    Web Actions choice can be obtained by invoking the advisor **Get Likely Web Action**, and its prediction model is updated by invoking the informant **Customer Action**.
- **Web Site Duration**
  - Web Site Durations's prediction model is updated by invoking the informant **Close Session**.
- **Web Support Types**
  - Web Support Types choice can be obtained by invoking the advisor **Get Web Support Types**, and its prediction model is updated by invoking the informant **Web Support Feedback**.
  - Returns likelihood for a customer to accept chat request or email support if offered.

4.2.3 Informants Usage with Offer-Oriented and Transaction-Oriented Choices

This section describes the following usages of the RTD_Base_ECommerce Inline Service informants:

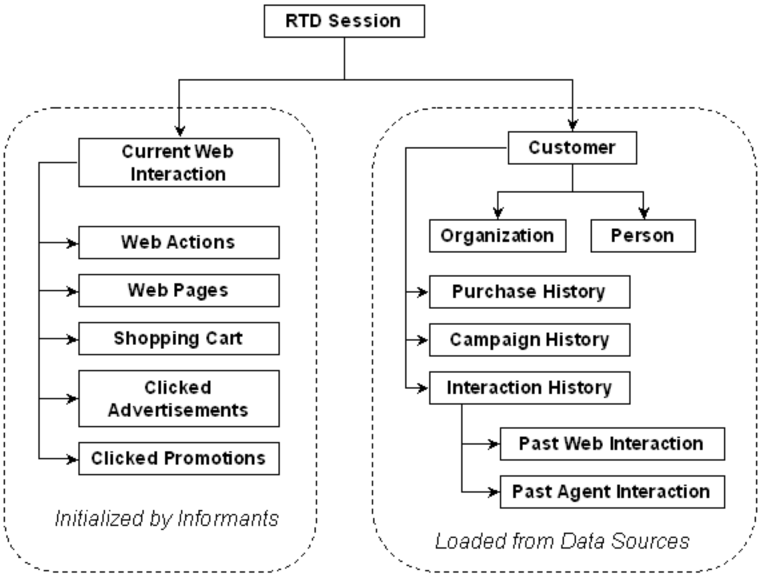
- How the informants provide feedback and additional information to Oracle RTD for the offer-oriented use cases described in Section 4.2.1, "Offer-Oriented Choices"
- How the informants are used for updating the choice groups described in Section 4.2.2, "Interaction-Oriented Choices"

For more specific information about the informants, see Section 4.4, "Informants."

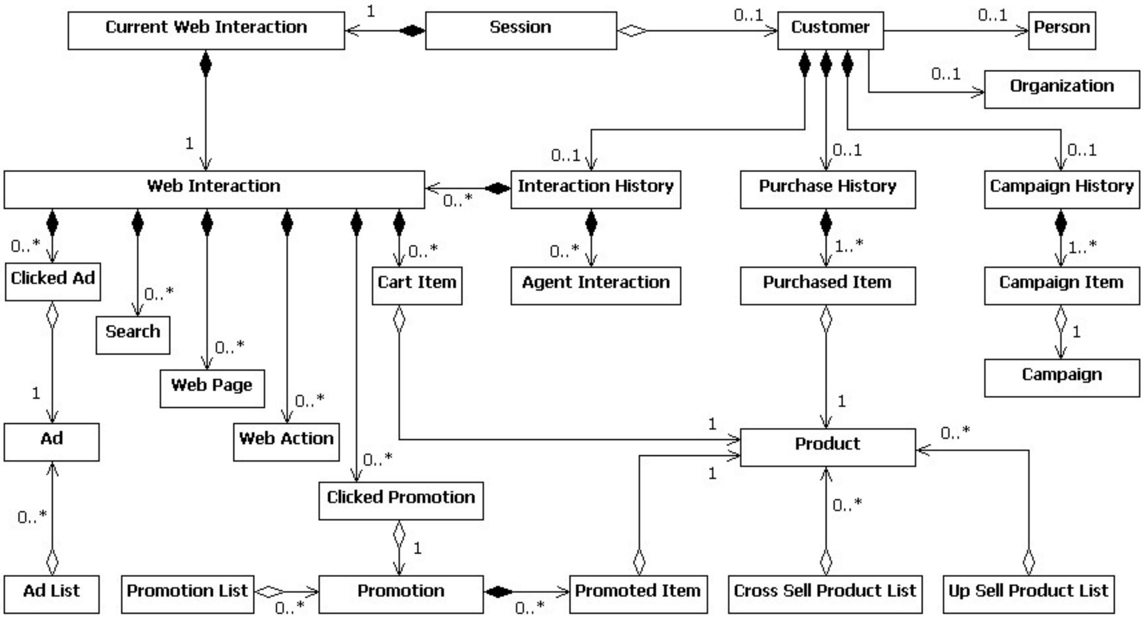
- **Initiate Session**
  - Updates current web interaction attributes for the session.
- **Identify Customer**
  - Updates customer related attributes for the session.
  - Provides Oracle RTD with customer id am profile information if a customer is identified while on the web site. In most cases, identification can be made upon user logon or by previously set cookies at the customer's browser.
- **Page Turn**
  - Updates current web interaction attributes and Web Page for the session.
  - Identifies any given page a customer navigates to that needs to be tracked.
- **Performed Search**
  - Updates current web interaction attributes and Search keywords for the session.
  - Identifies search words used by a customer.
- **Customer Action**
  - Updates current web interaction attributes and Web Action for the session.
  - Identifies key actions performed by a customer that can later be applied to entity attributes or models, for example, Update Customer Profile, Register, Un-register, Chat request.
- **Added To Cart**
  - Updates current web interaction attributes and Shopping Cart Items for the session.
- **Offer Response**
  - Creates prediction model entries and also captures shopping cart addition events.
  - Identifies if an offer (Upsell, Cross Sell, Promotion, or Ad) is clicked, added to a cart, or purchased by a web user or customer.
- **Close Session**
  - Creates analytical prediction model entries and triggers learning.
  - Formally closes out the Oracle RTD session.

4.3 Entities

The following diagram illustrates the overall general logical entity object model of the RTD_Base_ECommerce Inline Service.



The following diagram shows the entities and the relationships between the entities defined in the RTD_Base_ECommerce Inline Service.



Notation

The diagram shows the standard notation used in UML class diagrams, with each directed line representing a relationship *from* element A *to* element B, as follows:

- A line with no diamond represents an **association** between elements A and B.
- A line with a clear diamond represents an **aggregation**, with element A "owning" element B. However, with aggregation, removing an instance of element A does not remove the corresponding B element instances.

- A line with a filled diamond represents a **composition**, with element A "owning" element B. For a composition, removing an instance of element A also removes the corresponding B element instances.

The multiplicity of a relationship restricts how many element B instances the relationship may have. The restriction denotes either a precise limit, such as 1 or 0..1, or an open-ended upper limit, such as "zero or more" or "one or more."

For example:

- A Web Interaction may have many Web Pages. Deleting a particular Web Interaction automatically deletes the corresponding Web Pages.
- A Cart Item has one Product. Deleting the Cart Item does not delete the corresponding Product.

System-Oriented Entities

The Session entity is a built-in Oracle RTD entity for maintaining session attribute values.

The following entities are used for dynamic choice retrieval from external data sources: Ad List, Promotion List, Cross Sell Product List, and Up Sell Product List.

Entities Outline

The Current Web Interaction entity is a session attribute that keeps track of the current user interaction with the client system.

The Current Web Interaction entity references the Web Interaction entity, that itself keeps track of the following data:

- Clicked advertisements, using the Clicked Ad entity
- Clicked promotions, using the Clicked Promotion entity
- Searches performed, using the Search entity
- Actions performed, using the Web Action entity
- Visited Web pages, using the Web Page entity
- Items added to the shopping cart, using the Cart Item entity

The Customer entity is a session attribute that contains details of the customer profile as well as past customer interaction behavior.

A customer can be a Person or an Organization.

The Customer may have historical information, in the entities Purchase History, Campaign History, and Interaction History. Past customer interactions could be either or both of the following:

- Self-service Web-based interactions, using the Web Interaction entity
- Interactions through agents, using the Agent Interaction entity

After a Customer Id has been identified and supplied to Oracle RTD by the Identify Customer informant, Oracle RTD retrieves data from external data sources for the Customer entity and for its associated entity attributes.

In addition to the entity attributes that are normally mapped to a data source directly, derived attributes are also included, which obtain their values via Java functions that utilize the applicable raw data extracted from data sources as inputs.

This section describes the following entities:

- Session Entity
- Ad
- Ad List
- Agent Interaction
- Campaign
- Campaign History
- Campaign Item
- Cart Item
- Clicked Ad
- Clicked Promotion
- Cross Sell Product List
- Current Web Interaction
- Customer
- Interaction History
- Organization
- Person
- Product
- Promoted Item
- Promotion
- Promotion List
- Purchased Item
- Purchase History
- Rank Offers
- Search
- Up Sell Product List
- Web Action
- Web Interaction
- Web Page

4.3.1 Session Entity

(Key = Web Session Id)

Table 4–1 Session Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Customer	No	Customer	Yes	Yes	None
Current Web Interaction	No	Current Web Interaction	Yes	Yes	None

Table 4–1 (Cont.) Session Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Rank Offers	No	Rank Offers	No	No	Utilized when ranking offers directly from an Offer advisor instead of read from a data source.
Supplied Product Id	No	String	No	No	Utilized when ranking offers directly from an Offer advisor instead of read from a data source.

4.3.2 Ad

The Ad entity is used in conjunction with the Ad List entity and is used for the dynamic choice associated with the Ad Choice group.

Table 4–2 Ad Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Ad Id	No	String	No	No	Key
Category	No	String	No	No	None
Name	No	String	No	No	None
Type	No	String	No	No	None

4.3.3 Ad List

The Ad List entity is used in conjunction with the Ad entity and is used for the dynamic choice associated with the Ad Choice group.

Table 4–3 Ad List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Active	No	String	No	No	Key
Ads	Yes	Ad	No	No	The Ads attribute is an array attribute based on the Ad entity.

4.3.4 Agent Interaction

The Agent Interaction entity contains attributes related to agent interactions that have taken place with a customer. This entity is used to create an array of agent interactions within the Interaction History entity, which in turn is associated with the Customer entity.

Table 4–4 Agent Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction Id	No	String	No	No	None
Agent Id	No	String	No	No	None
Agent Location	No	String	No	No	None
Customer Location	No	String	No	No	None
Interaction Channel	No	String	No	No	None
Interaction Date	No	None	No	No	None
Interaction Duration	No	Integer	No	No	None
Interaction Reason	No	String	No	No	None
Interaction Status	No	String	No	No	None
Interaction Type	No	String	No	No	None
Time Of Day	No	String	No	No	None

4.3.5 Campaign

The Campaign entity contains attributes related to campaigns that have been associated with a customer. This entity is associated with the customer via the Campaign Item and Campaign History entities.

Table 4–5 Campaign Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Campaign Id	No	String	No	No	None
Category	No	String	Yes	Yes	None
Name	No	String	Yes	Yes	None
Period	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None

4.3.6 Campaign History

The Campaign History entity contains attributes related to campaigns that have been associated with a customer. This entity is associated with the Customer entity and also contains the Campaign Items entity.

Table 4–6 Campaign History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Last Campaign Category	No	String	Yes	Yes	Default Value based on the Get Last Campaign function
Last Campaign Date	No	Date	Yes	Yes	Default Value based on the Get Last Campaign Date function

Table 4–6 (Cont.) Campaign History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Last Campaign Delivery Method	No	String	Yes	Yes	Default Value based on the Get Last Campaign Delivery Method function
Last Campaign Name	No	String	Yes	Yes	Default Value based on the Get Last Campaign Name function
Last Campaign Type	No	String	Yes	Yes	Default Value based on the Get Last Campaign Type function
Days Since Last Campaign	No	Integer	Yes	Yes	Default Value based on the Get Days Since Last Campaign
Campaign Items	Yes	Campaign Item	Yes	Yes	The Campaign Items attribute is an array attribute based on the Campaign Item entity.

4.3.7 Campaign Item

The Campaign Item entity contains attributes related to campaigns that have been associated with a customer. This entity is associated with the customer via the Campaign History entity and contains the Campaign entity.

Table 4–7 Campaign Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Campaign Date	No	Date	Yes	Yes	None
Delivery Method	No	String	Yes	Yes	None
Campaign	No	Campaign	Yes	Yes	The Campaign attribute is based on the Campaign entity.

4.3.8 Cart Item

The Cart Item entity contains attributes related to the products that a customer has put in their shopping cart in their web session. The Cart Item entity contains the Product entity, and is itself embedded in the Web Interaction entity.

Table 4–8 Cart Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Quantity	No	Integer	Yes	Yes	None
Product	No	Product	Yes	Yes	None

4.3.9 Clicked Ad

The Clicked Ad entity contains attributes related to the ads which a customer may have clicked during their web session. The Clicked Ad entity contains the Ad entity, and is itself embedded in the Web Interaction entity.

Table 4–9 Clicked Ad Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Count	No	Integer	Yes	Yes	None
Ad	No	Ad	Yes	Yes	None

4.3.10 Clicked Promotion

The Clicked Promotion entity contains attributes related to the ads which a customer may have clicked during their web session. The Clicked Promotion entity contains the Promotion entity, and is itself embedded in the Web Interaction entity.

Table 4–10 Clicked Promotion Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Count	No	Integer	Yes	Yes	None
Promotion	No	Promotion	Yes	Yes	None

4.3.11 Cross Sell Product List

The Cross Sell Product List entity is used in conjunction with the Products entity (instantiated as Cross Sell Products), and is used for the dynamic choice associated with the Cross Sell Offers Choice group.

Table 4–11 Cross Sell Product List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Product Id	No	String	No	No	None
Cross Sell Products	Yes	Product	No	No	The Cross Sell Products attribute is an array attribute based on the Product entity.
Cross Sell Products - Product Id	No	String	No	No	None

4.3.12 Current Web Interaction

The Current Web Interaction entity contains attributes related to what a customer is doing during their current web session. The Current Web Interaction entity contains its own attributes as well as attributes from the Web Interaction entity.

See Section 4.3.27.1, "Derivation of Web Interaction Attributes in Referencing Entities" for details of how the Web Interaction attributes Interaction Date, Start Time, and Total Duration in Minutes are derived.

Table 4–12 Current Web Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction Id	No	String	Yes	Yes	None
Current Page	No	String	Yes	Yes	None
Current Page Type	No	String	Yes	Yes	None
Current Viewed Product Id	No	String	Yes	Yes	None
Last Added Product Id	No	String	Yes	Yes	None
Previous Page	No	String	Yes	Yes	None
Previous Page Type	No	String	Yes	Yes	None
Time Spent on Previous Page	No	Integer	Yes	Yes	None
Web Interaction	No	Web Interaction	Yes	Yes	None

4.3.13 Customer

The Customer entity contains attributes related to the profile of the customer. The Customer entity contains its own attributes and links in the Campaign History, Interaction History, Organization, Person, and Purchase History entities.

Table 4–13 Customer Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Customer Id	No	String	No	No	Key
Address City	No	String	Yes	Yes	None
Address Country	No	String	Yes	Yes	None
Address Postal Code	No	String	Yes	Yes	None
Address Region	No	String	Yes	Yes	None
Address State Province	No	String	Yes	Yes	None
Credit Hold	No	String	Yes	Yes	None
Life Time Value Score	No	Double	Yes	Yes	None
Offline Churn Propensity	No	Double	Yes	Yes	None
Phone Area Code	No	String	Yes	Yes	None
Preferred Language	No	String	Yes	Yes	None
Start Date	No	Date	Yes	Yes	None
Status	No	String	Yes	Yes	None
Target Market Segment	No	String	Yes	Yes	None
Tenure	No	Integer	Yes	Yes	None
Total Credit Limit	No	Double	Yes	Yes	None

Table 4–13 (Cont.) Customer Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Type	No	String	Yes	Yes	Default Value determined by Get Customer Type function.
Campaign History	No	Campaign History	Yes	Yes	The Campaign History attribute is based on the Campaign History entity. See Campaign History entity for attribute breakdown.
Interaction History	No	Interaction History	Yes	Yes	The Interaction History attribute is based on the Interaction History entity. See Interaction History entity for attribute breakdown.
Organization	No	Organization	Yes	Yes	The Organization attribute is based on the Organization entity. See Organization entity for attribute breakdown.
Person	No	Person	Yes	Yes	The Person attribute is based on the Person entity. See Person entity for attribute breakdown.
Purchase History	No	Purchase History	Yes	Yes	The Purchase History attribute is based on the Purchase History entity. See Purchase History entity for attribute breakdown.

4.3.14 Interaction History

The Interaction History entity contains attributes that record what a customer has done in the past regarding previous interactions. The Interaction History entity contains derived attributes from both previous Agent Interactions and previous Web Interactions.

See Section 4.3.27.1, "Derivation of Web Interaction Attributes in Referencing Entities" for details of how the Past Web Interactions attributes Interaction Date, Start Time, and Total Duration in Minutes are derived.

Table 4–14 Interaction History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Agent Interaction Reasons In Past 30 Days	Yes	String	Yes	Yes	Default determined by Get Agent Interaction Reasons In Past Days function
Agent Interaction Types in Past 30 Days	Yes	String	Yes	Yes	Default determined by Get Agent Interaction Types In Past Days function
Days Since Last Agent Interaction	No	Integer	Yes	Yes	Default determined by Get Days Since Last Agent Interaction function
Days Since Last Interaction	No	Integer	Yes	Yes	Default determined by Maximum function
Days Since Last Web Interaction	No	Integer	Yes	Yes	Default determined by Get Days Since Last Web Interaction function
Interaction Types In Past 30 Days	No	String	Yes	Yes	Default determined by Get Interaction Types function
Last Agent Interaction Status	No	String	Yes	Yes	Default determined by Get Last Agent Interaction Status function
Last Agent Interaction Type	No	String	Yes	Yes	Default determined by Get Last Agent Interaction Type function
Last Interaction Type	No	String	Yes	Yes	Default determined by Get Last Interaction Type function
Number of Agent Interactions In Past 30 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Agent Interaction in Past Days function
Number of Agent Interactions In Past 90 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Agent Interaction in Past Days function
Number of Agent Interactions In Past Year	No	Integer	Yes	Yes	Default determined by Get Number Of Agent Interaction in Past Days function
Number of Web Interactions In Past 30 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Web Interaction in Past Days function

Table 4–14 (Cont.) Interaction History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Number of Web Interactions In Past 90 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Web Interaction in Past Days function
Number of Web Interactions In Past Year	No	Integer	Yes	Yes	Default determined by Get Number Of Web Interaction in Past Days function
Performed Web Actions In Past 30 Days	Yes	String	Yes	Yes	Default determined by Get Performed Web Actions in Past Days function
Past Agent Interactions	Yes	Agent Interaction	No	No	None
Past Web Interactions	Yes	Web Interaction	No	No	None

4.3.15 Organization

The Organization entity contains attributes related to the profile of an Organization. The Organization entity is linked to the session through the Customer entity.

Table 4–15 Organization Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Annual Gross Profit	No	Double	Yes	Yes	None
Annual Revenue	No	Double	Yes	Yes	None
Business Partner Flag	No	String	Yes	Yes	None
Established Service	No	Integer	Yes	Yes	None
Line Of Business	No	String	Yes	Yes	None
Number of Employees	No	Integer	Yes	Yes	None
Number Of Years Established	No	Integer	Yes	Yes	None
Size	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None

4.3.16 Person

The Person entity contains attributes related to the profile of a Person. The Person entity is linked to the session through the Customer entity.

Table 4–16 Person Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Age	No	Integer	Yes	Yes	None
Annual Income	No	Double	Yes	Yes	None

Table 4–16 (Cont.) Person Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Credit Score	No	Integer	Yes	Yes	None
Education Level	No	String	Yes	Yes	None
Ethnicity	No	String	Yes	Yes	None
Gender	No	String	Yes	Yes	None
Marital Status	No	String	Yes	Yes	None
Net Worth	No	Double	Yes	Yes	None
Number Of Children	No	Integer	Yes	Yes	None
Profession	No	String	Yes	Yes	None

4.3.17 Product

The Product entity contains attributes related to a generic Product. The Product entity is used as a reference entity under the Cart Item, Cross Sell Product List, Promoted Item, Purchased Item, and Up Sell Product entities.

Table 4–17 Product Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Product Id	No	String	No	No	Key
Category	No	String	Yes	Yes	None
Name	No	String	Yes	Yes	None
Popularity Rank	No	Integer	Yes	Yes	None
Product Line	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None
Unit Price	No	Double	Yes	Yes	None

4.3.18 Promoted Item

The Promoted Item entity contains attributes related to a Promoted Item. The Promoted Item entity is used in conjunction with the Promotion entity, which can contain multiple promoted Items.

Table 4–18 Promoted Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Discount Rate	No	Double	Yes	Yes	None
Promoted Product	No	Product	Yes	Yes	None

4.3.19 Promotion

The Promotion entity is used in conjunction with the Promotion List entity, and is used for the dynamic choice associated with the Promotions Choice group.

Table 4–19 Promotion Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Promotion Id	No	String	No	No	None
Category	No	String	No	No	None
Days Left	No	Integer	No	No	Default Value set by Get Days Left function
Duration In Days	No	Integer	No	No	Default Value set by Get Duration In Days function
Effective Date	No	Date	No	No	None
Expiry Date	No	Date	No	No	None
Name	No	String	No	No	None
Period	No	String	No	No	None
Type	No	String	No	No	None
Promoted Items	Yes	Promoted Item	No	No	The Promoted Items attribute is based on the Promoted Item entity. See Promoted Item entity for attribute breakdown.

4.3.20 Promotion List

The Promotion List entity is used in conjunction with the Promotion entity, and is used for the dynamic choice associated with the Promotions Choice group.

Table 4–20 Promotion List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Active	No	String	No	No	Key
Promotions	Yes	Promotion	No	No	The Promotions attribute is based on the Promotion entity. See Promotion entity for attribute breakdown.

4.3.21 Purchased Item

The Purchased Item entity contains attributes related to a purchased item. This entity is associated with the Customer entity through the Purchase History entity.

Table 4–21 Purchased Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Purchase Amount	No	Double	Yes	Yes	None
Purchase Date	No	Date	Yes	Yes	None
Purchased Product	No	Product	Yes	Yes	None

4.3.22 Purchase History

The Purchased History entity contains attributes related to a customer's past purchases. This entity is associated with the session via the Customer entity.

Table 4–22 Purchase History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Days Since Last Purchase	No	Integer	Yes	Yes	Default Value set by Get Days Since Last Purchase function
Last Purchase Amount	No	Double	Yes	Yes	Default Value set by Get Last Purchase Amount function
Last Purchased Product	No	String	Yes	Yes	Default Value set by Get Last Purchased Product function
Last Purchased Product Line	No	String	Yes	Yes	Default Value set by Get Last Purchased Product Line function
Product Lines Owned	Yes	String	Yes	Yes	Default Value set by Get Product Lines Owned function
Total Amount Spent	No	Double	Yes	Yes	Default Value set by Get Total Amount Spent function
Total Amount Spent in Last 90 Days	No	Double	Yes	Yes	Default Value set by Get Total Amount Spent In Last 90 Days function
Purchased Items	Yes	Purchased Item	Yes	Yes	The Purchased Items array attribute is based on the Purchased Item entity. See Purchased Item entity for attribute breakdown.

4.3.23 Rank Offers

The Rank Offers entity is used to store arrays of different offer types that are passed to the Inline Service via an advisor. After it is filled by the advisor input, the entity is then used to populate the corresponding dynamic choice.

Table 4–23 Rank Offers Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Ads	Yes	String	No	No	None
CrossSellOffers	Yes	String	No	No	None
In Use	No	Boolean	No	No	Default Value is False
Promotions	Yes	String	No	No	
UpSellOffers	Yes	String	No	No	

4.3.24 Search

The Search entity is used to store attributes related to search strings performed by a customer. This entity is used by the Web Interaction entity as an array attribute.

Table 4–24 Search Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Count	No	Integer	Yes	Yes	None
Keyword	No	String	Yes	Yes	None

4.3.25 Up Sell Product List

The Up Sell Product List entity is used in conjunction with the Products entity (instantiated as Up Sell Products), and is used for the dynamic choice associated with the Up Sell Offers Choice group.

Table 4–25 Up Sell Product List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Product Id	No	String	No	No	None
Up Sell Products	Yes	Product	No	No	The Up Sell Products attribute is an array attribute based on the Product entity.

4.3.26 Web Action

The Web Action entity is used to store attributes related to actions performed by a customer while on the web. This entity is used by the Web Interaction entity as an array attribute.

Table 4–26 Web Action Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Name	No	String	No	No	Key
Category	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None
Web Action Id	No	String	Yes	Yes	None

4.3.27 Web Interaction

The Web Interaction entity contains attributes related to a generic Web Interaction. This entity is used as a reference entity under for the Current Web Interaction entity as well as an array attribute (Past Web Interactions) under the Interaction History entity.

See Section 4.3.27.1, "Derivation of Web Interaction Attributes in Referencing Entities" for details of how the Interaction Date, Start Time, and Total Duration in Minutes attributes are used in the Current Web Interaction entity and the Interaction History entity.

Table 4–27 Web Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction Id	No	String	No	No	Key
Clicked Ads	Yes	Clicked Ad	Yes	Yes	The Clicked Ads attribute is an array attribute based on the Clicked Ad entity. See the Clicked Ad entity for attribute breakdown.
Clicked Promotions	Yes	Clicked Promotion	Yes	Yes	The Clicked Promotions attribute is an array attribute based on the Clicked Promotion entity. See the Clicked Promotion entity for attribute breakdown.

Table 4–27 (Cont.) Web Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction Date	No	Date	Yes	Yes	None
Origin To Website	No	String	Yes	Yes	None
Performed Actions	Yes	Web Action	Yes	Yes	The Performed Actions attribute is an array attribute based on the Web Action entity. See the Web Action entity for attribute breakdown.
Performed Searches	Yes	Search	Yes	Yes	The Performed Searches attribute is an array attribute based on the Search entity. See the Search entity for attribute breakdown.
Shopping Cart Items	Yes	Cart Item	Yes	Yes	The Shopping Cart Items attribute is an array attribute based on the Cart Item entity. See the Cart Item entity for attribute breakdown.
Start Time	No	Date	No	No	Attribute used to derive the Web Interaction attribute Total Duration in Minutes when used in the Current Web Interaction entity.
Time Of Day	No	String	Yes	Yes	None
Total Duration in Minutes	No	String	Yes	Yes	None
Visited Pages	Yes	Web Page	Yes	Yes	The Visited Pages attribute is an array attribute based on the Web Page entity. See the Web Page entity for attribute breakdown.
Web User Location	No	String	Yes	Yes	None

4.3.27.1 Derivation of Web Interaction Attributes in Referencing Entities

When the Current Web Interaction and the Interaction History reference the Web Interaction entity, special considerations apply to the derivation of certain attributes, as follows:

Table 4–28 Derivation of Web Interaction Attributes in Referencing Entities

Web Interaction Attribute	As used in the Current Web Interaction Entity attribute Web Interaction	As used in the Interaction History Entity attribute Past Web Interactions
Interaction Date	NA	Retrieved from the external data source using Web Interaction mapping
Start Date	Initialized by informant Initiate Session with the Current Time	Not initialized
Total Duration in Minutes	Computed as the difference between Web Interaction-Start Time and Current Time	Retrieved from the external data source using Web Interaction mapping

4.3.28 Web Page

The Web Page entity is used to store attributes related to the attributes of a web page that a customer has visited while on the web. This entity is used by the Web Interaction entity as an array attribute.

Table 4–29 Web Page Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Web Page Id	No	String	No	No	Key
Average Time Spent On Page	No	Double	Yes	Yes	None
Category	No	String	Yes	Yes	None
Count	No	Integer	Yes	Yes	None
Name	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None

4.4 Informants

This section describes the following Informants:

- Initiate Session
- Identify Customer
- Page Turn
- Performed Search
- Customer Action
- Added To Cart
- Offer Response
- Web Support Feedback
- Close Session

4.4.1 Initiate Session

The informant Initiate Session creates the session for the interaction and updates current web interaction attributes for the session.

Table 4–30 describes the parameters for the informant Initiate Session.

Table 4–30 Informant Initiate Session

Parameter	Description
Informant Name	Initiate Session
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Origin To Website => Session / Current Web Interaction / Web Interaction / Origin To Website Time Of Day => Session / Current Web Interaction / Web Interaction / Time Of Day User Location => Session / Current Web Interaction / Web Interaction / Web User Location

Table 4–30 (Cont.) Informant Initiate Session

Parameter	Description
External System	Web E-Commerce
Order	0
Force session close	No
Logic	None
Pre-condition	None

4.4.2 Identify Customer

The informant Identify Customer updates and triggers loading of the following customer related attributes for the session:

- Customer Profile
- Interaction History
- Purchase History
- Campaign History

The caller provides Oracle RTD with the Customer Id after a customer is identified while on the web site. In most cases, identification can be made upon user logon or by previously set cookies on the caller's web browser.

Table 4–31 describes the parameters for the informant Identify Customer.

Table 4–31 Informant Identify Customer

Parameter	Description
Informant Name	Identify Customer
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Customer Id => Customer / Customer Id
External System	Web E-Commerce
Order	0
Force session close	No
Logic	None
Pre-condition	None

4.4.3 Page Turn

The informant Page Turn updates current web interaction attributes (Web Pages) for the session. Identifies any given page a customer navigates to that needs to be tracked

Table 4–32 describes the parameters for the informant Page Turn.

Table 4–32 Informant Page Turn

Parameter	Description
Informant Name	Page Turn
Session Keys	Session / Current Web Interaction / Interaction Id

Table 4–32 (Cont.) Informant Page Turn

Parameter	Description
Request Data	Current Page => Session / Current Web Interaction / Current Page Current Page Type => Session / Current Web Interaction / Current Page Type Previous Page => Session / Current Web Interaction / Previous Page Previous Page Type => Session / Current Web Interaction / Previous Page Type Time Spent on Previous Page => Session / Current Web Interaction / Time Spent on Previous Page Current Viewed Product Id => Session / Current Web Interaction / Current Viewed Product Id
External System	Web E-Commerce
Order	0
Force session close	No
Logic	(Asynchronous) Update the nested Web Pages session attributes
Pre-condition	None

Supplied Current Viewed Product Id can be used later as the base product id in making cross sell or up sell decision.

4.4.4 Performed Search

The informant Performed Search updates current web interaction attributes (Search Keywords) for the session. It identifies search words used by a user or customer.

Table 4–33 describes the parameters for the informant Performed Search.

Table 4–33 Informant Performed Search

Parameter	Description
Informant Name	Performed Search
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Search Keyword
External System	Web E-Commerce
Order	0
Force session close	No
Logic	(Asynchronous) Update the nested Searches session attributes
Pre-condition	None

4.4.5 Customer Action

The informant Customer Action updates current web interaction attributes (Web Action) for the session. It identifies key actions performed by a customer that can later be applied to models, for example, Update Customer Profile, Register, Un-register, Chat request.

Table 4–34 describes the parameters for the informant Customer Action.

Table 4–34 Informant Customer Action

Parameter	Description
Informant Name	Customer Action
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Action Name
External System	Web E-Commerce
Order	0
Force session close	No
Logic	Update Web Action Model (choice model) Create or Update Web Action session variable of current interaction
Pre-condition	None

Set Action Name as a choice name into the Web Action Model.

4.4.6 Added To Cart

The informant Added To Cart updates current web interaction attributes (Cart Item) for the session. It registers a shopping cart item along with its quantity added during the session.

Table 4–35 describes the parameters for the informant Added To Cart.

Table 4–35 Informant Added to Cart

Parameter	Description
Informant Name	Customer Action
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id Quantity
External System	Web E-Commerce
Order	0
Force session close	No
Logic	Update Web Action Model (choice model) Create or Update Shopping Cart Item session variable of current interaction
Pre-condition	None

4.4.7 Offer Response

The informant Offer Response creates prediction model entries and also captures shopping cart addition events. It identifies if an offer (Up Sell, Cross Sell, Promotion, Ad) is clicked, added to cart, or purchased by a web user or customer.

Table 4–36 describes the parameters for the informant Offer Response.

Table 4–36 Informant Offer Response

Parameter	Description
Informant Name	Offer Response
Session Keys	Session / Current Web Interaction / Interaction Id

Table 4–36 (Cont.) Informant Offer Response

Parameter	Description

Request Data	Offer Id {choice id}
	Offer Type {Up Sell, Cross Sell, Promotion, Ad}
	Event {Interested, Added To Cart, Purchased}

External System	Web E-Commerce
Order	0
Force session close	No
Logic	If event = Interested then if offer type = Promotion then create/update Clicked Promotion (increase count if previously exist) if offer type = Ad then create/update Clicked Ad (increase count if previously exist) Record the event into the appropriate model for the choice (determined by choice id / offer id)
Pre-condition	None

4.4.8 Web Support Feedback

The informant Web Support Feedback updates the Web Support prediction model.

Table 4–37 describes the parameters for the informant Web Support Feedback.

Table 4–37 Informant Web Support Feedback

Parameter	Description
Informant Name	Web Support Feedback
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Choice Name Choice Event
External System	Web E-Commerce
Order	0
Force session close	No
Logic	If Choice Event = Used then record the event into the Web Support Usage Model
Pre-condition	None

Dependencies:

- Function - Set Choice Event Model (Model Name, Choice Name, Choice Event)

4.4.9 Close Session

The informant Close Session creates analytical prediction model entries, triggers learning, and formally closes out the Oracle RTD session.

Table 4–38 describes the parameters for the informant Close Session.

Table 4–38 Informant Close Session

Parameter	Description
Informant Name	Close Session
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Abandonment Flag Interaction Duration
External System	Web E-Commerce
Order	0
Force session close	No
Logic	Update Abandonment Model. Possible values for Abandonment Flag are: "Abandoned" "Not Abandoned" Update Web Site Duration Model. Possible values for Interaction Duration are: "00 to 05 Minutes" "05 to 10 Minutes" "10 to 15 Minutes" "15 to 20 Minutes" "20 Minutes or Greater"
Pre-condition	None

Dependencies:

- Function - Set Choice Model (Model Name, Choice Name)

4.5 Advisors and Associated Decisions, Choice Groups, and Models

For each advisor listed in this section, a detailed breakdown is provided for the Integration Point, followed by:

- The decision called by the advisor
- The choice group used by the decision
- As appropriate, the model associated with the choice group

This section consists of the following topics:

- Section 4.5.1, "Decisioning Advisors"
- Section 4.5.2, "Analysis Advisors"

4.5.1 Decisioning Advisors

This section describes the following advisors:

- Get Upsell Offers
- Get Cross Sell Offers
- Get Promotions

- Get Advertisements
- Get Offers

4.5.1.1 Get Upsell Offers

The advisor Get Upsell Offers determines the likelihood for a customer to accept an upsell offer.

Table 4–39 describes the parameters for the advisor Get Upsell Offers.

Table 4–39 Advisor Get Upsell Offers

Parameter	Description
Advisor Name	Get Upsell Offers
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id (optional) Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Upsell Offers
Group Decision	Select Upsell Offers
Default Choices	None
Logic	The supplied product id is assigned to Session / Supplied Product Id. This will be used by function: Select Product Id (for Cross Sell / Up Sell). If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	Base Product Id for Cross sell / Up sell must exist. Function: Select Product Id tries to find this Product Id from the following variables: 1 - Session / Supplied Product Id 2 - Session / Current Web Interaction / Current Viewed Product Id 3 - Session / Current Web Interaction / Last Added (into shopping cart) Product Id

Table 4–40 describes the parameters for the decision for the advisor Get Upsell Offers.

Table 4–40 Decision for Advisor Get Upsell Offers

Parameter	Description
Decision Name	Select Upsell Offers
Select Choices From	Upsell Offers
Number of Choices to Select	5
Select at Random	No

Table 4–40 (Cont.) Decision for Advisor Get Upsell Offers

Parameter	Description
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 33% Maximize Revenue 33% Popularity 33%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 4–41 describes the parameters for the choice group for the advisor Get Upsell Offers.

Table 4–41 Choice Group for Advisor Get Upsell Offers

Parameter	Description
Choice Group Name	Upsell Offers
Choice Attributes	Product Id Product Name Product Line Product Type Product Category Offer Type = "Up Sell" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Upsell Purchase Model: Purchased Maximize Revenue = Dynamic Choice / Unit Price Popularity = Dynamic Choice / Popularity Rank
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers) Added To Cart Purchased
Choice Eligibility	None
Group Attributes	Upsell Products – Type=Product (Array) – Loading: Get Up Sell Product List (Select Product Id())
Group Eligibility	None
Dynamic Choices	Choice Id is Product Id

Table 4–42 describes the parameters for the model for the advisor Get Upsell Offers.

Table 4–42 Model for the Advisor Get Upsell Offers

Parameter	Description
Model Name	Upsell Purchase Model (choice event model)

Table 4–42 (Cont.) Model for the Advisor Get Upsell Offers

Parameter	Description
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Upsell Offers
Base Event	Presented
Positive Events	Purchased
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Offer Response Get Upsell Offers Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers
- Function - Select Product Id ( )
- Function - Get Up Sell Product List (Product Id)

4.5.1.2 Get Cross Sell Offers

The advisor Get Cross Sell Offers determines the likelihood for a customer to accept a cross sell offer.

Table 4–43 describes the parameters for the advisor Get Cross Sell Offers.

Table 4–43 Advisor Get Cross Sell Offers

Parameter	Description
Advisor Name	Get Cross Sell Offers
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id (optional) Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)
External System	Web E-Commerce
Order	0

Table 4–43 (Cont.) Advisor Get Cross Sell Offers

Parameter	Description
Force session close	No
Decision	Select Cross Sell Offers
Group Decision	Select Cross Sell Offers
Default Choices	None
Logic	The supplied product id is assigned to Session / Supplied Product Id. This will be used by function: Select Product Id (for Cross Sell / Up Sell). If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	Base Product Id for Cross sell / Up sell must exist. Function: Select Product Id tries to find this Product Id from the following variables: 1 - Session / Supplied Product Id 2 - Session / Current Web Interaction / Current Viewed Product Id 3 - Session / Current Web Interaction / Last Added (into shopping cart) Product Id

Table 4–44 describes the parameters for the decision for the advisor Get Cross Sell Offers.

Table 4–44 Decision for Advisor Get Cross Sell Offers

Parameter	Description
Decision Name	Select Cross Sell Offers
Select Choices From	Cross Sell Offers
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 33% Maximize Revenue 33% Popularity 33%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 4–45 describes the parameters for the choice group for the advisor Get Cross Sell Offers.

Table 4–45 Choice Group for Advisor Get Cross Sell Offers

Parameter	Description
Choice Group Name	Cross Sell Offers

Table 4–45 (Cont.) Choice Group for Advisor Get Cross Sell Offers

Parameter	Description
Choice Attributes	Product Id Product Name Product Line Product Type Product Category Offer Type = "Cross Sell" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Cross Sell Purchase Model: Purchased Maximize Revenue = Dynamic Choice / Unit Price Popularity = Dynamic Choice / Popularity Rank
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers) Added To Cart Purchased
Choice Eligibility	None
Group Attributes	Cross Sell Products - Type=Product (Array) - Loading: Get Cross Sell Product List (Select Product Id())
Group Eligibility	None
Dynamic Choices	Choice Id is Product Id

Table 4–46 describes the parameters for the model for the advisor Get Cross Sell Offers.

Table 4–46 Model for Advisor Get Cross Sell Offers

Parameter	Description
Model Name	Cross Sell Purchase Model (choice event model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Cross Sell Offers
Base Event	Presented
Positive Events	Purchased
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Offer Response Get Cross Sell Offers Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer

- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers
- Function - Select Product Id ( )
- Function - Get Cross Sell Product List (Product Id)

4.5.1.3 Get Promotions

The advisor Get Promotions determines the likelihood for a customer to have an interest on a presented promotion.

Table 4–47 describes the parameters for the advisor Get Promotions.

Table 4–47 Advisor Get Promotions

Parameter	Description
Advisor Name	Get Promotions
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Promotions
Group Decision	Select Promotions
Default Choices	None
Logic	If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	None

Table 4–48 describes the parameters for the decision for the advisor Get Promotions.

Table 4–48 Decision for Advisor Promotions

Parameter	Description
Decision Name	Select Promotions
Select Choices From	Promotions
Number of Choices to Select	5
Select at Random	No

Table 4–48 (Cont.) Decision for Advisor Promotions

Parameter	Description
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 4–49 describes the parameters for the choice group for the advisor Get Promotions.

Table 4–49 Choice Group for Advisor Promotions

Parameter	Description
Choice Group Name	Promotions
Choice Attributes	Promotion Id Promotion Name Promotion Type Promotion Period Offer Type = "Promotion" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Promotion Interest Model: Interested
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers)
Choice Eligibility	None
Group Attributes	Promotion List - Type=Promotion (Array) - Data loading function: Get Promotion List ()
Group Eligibility	None
Dynamic Choices	Choice Id is Promotion Id

Table 4–50 describes the parameters for the model for the advisor Get Promotions.

Table 4–50 Model for Advisor Get Promotions

Parameter	Description
Model Name	Promotion Interest Model (choice event model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Promotions
Base Event	Presented
Positive Events	Interested
Partitioning Attributes	None
Excluded Attributes	None

Table 4–50 (Cont.) Model for Advisor Get Promotions

Parameter	Description
Learn Location	Offer Response Get Promotions Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers

4.5.1.4 Get Advertisements

The advisor Get Advertisements determines the likelihood for a customer to have an interest on a presented advertisement.

Table 4–51 describes the parameters for the advisor Get Advertisements.

Table 4–51 Advisor Get Advertisements

Parameter	Description
Advisor Name	Get Advertisements
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Advertisements
Group Decision	Select Advertisements
Default Choices	None
Logic	If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	None

Table 4–52 describes the parameters for the decision for the advisor Get Advertisements.

Table 4–52 Decision for Advisor Get Advertisements

Parameter	Description
Decision Name	Select Advertisements
Select Choices From	Advertisements
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 4–53 describes the parameters for the choice group for the advisor Get Advertisements.

Table 4–53 Choice Group for Advisor Get Advertisements

Parameter	Description
Choice Group Name	Advertisements
Choice Attributes	Ad Id Ad Name Ad Type Ad Category Offer Type = "Ad" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Advertisement Interest Model: Interested
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers)
Choice Eligibility	None
Group Attributes	Ad List - Type=Ad (Array) - Data loading function: Get Ad List ()
Group Eligibility	None
Dynamic Choices	Choice Id is Ad Id

Table 4–54 describes the parameters for the model for the advisor Get Advertisements.

Table 4–54 Model for Advisor Get Advertisements

Parameter	Description
Model Name	Advertisement Interest Model (choice event model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian

Table 4–54 (Cont.) Model for Advisor Get Advertisements

Parameter	Description
Choice Group	Promotions
Base Event	Presented
Positive Events	Interested
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Offer Response Get Advertisements Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers

4.5.1.5 Get Offers

The advisor Get Offers determines the likelihood for a customer to accept an offer.

The advisor Get Offers returns a mix of offers:

- Upsell Offers
- Cross Sell Offers
- Promotions
- Advertisements

Table 4–55 describes the parameters for the advisor Get Offers.

Table 4–55 Advisor Get Offers

Parameter	Description
Advisor Name	Get Offers
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id (optional) Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)

Table 4–55 (Cont.) Advisor Get Offers

Parameter	Description
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Offers
Group Decision	Select Offers
Default Choices	None
Logic	The supplied product id is assigned to Session / Supplied Product Id. This will be used by function: Select Product Id (for Cross Sell / Up Sell). If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	Base Product Id for Cross sell / Up sell must exist. Function: Select Product Id tries to find this Product Id from the following variables: 1 - Session / Supplied Product Id 2 - Session / Current Web Interaction / Current Viewed Product Id 3 - Session / Current Web Interaction / Last Added (into shopping cart) Product Id

Table 4–56 describes the parameters for the decision for the advisor Get Offers.

Table 4–56 Decision for Advisor Get Offers

Parameter	Description
Decision Name	Select Offers
Select Choices From	Offers
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("presented");}

Table 4–57 describes the parameters for the choice group for the advisor Get Offers.

Table 4–57 Choice Group for Advisor Get Offers

Parameter	Description
Choice Group Name	Offers
Choice Attributes	Offer Type Likelihood

Table 4–57 (Cont.) Choice Group for Advisor Get Offers

Parameter	Description
Scores	Maximize Acceptance Likelihood
Choice Events	Presented Interested
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Upsell Offers
- Choice Group - Cross Sell Offers
- Choice Group - Promotions
- Choice Group - Advertisements

4.5.2 Analysis Advisors

This section describes the following advisors:

- Get Abandonment Propensity
- Get Likely Web Action
- Get Likely Web Duration
- Get Web Support Types

4.5.2.1 Get Abandonment Propensity

The advisor Get Abandonment Propensity returns the likelihood of abandonment for a customer.

Table 4–58 describes the parameters for the advisor Get Abandonment Propensity.

Table 4–58 Advisor Get Abandonment Propensity

Parameter	Description
Advisor Name	Get Abandonment Propensity
Session Keys	Session / Current Web Interaction / Interaction Id

Table 4–58 (Cont.) Advisor Get Abandonment Propensity

Parameter	Description
Request Data	None
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Abandonment Propensity
Group Decision	Select Abandonment Propensity
Default Choices	None
Logic	None
Pre-condition	None

Table 4–59 describes the parameters for the decision for the advisor Get Abandonment Propensity.

Table 4–59 Decision for Advisor Get Abandonment Propensity

Parameter	Description
Decision Name	Select Abandonment Propensity
Select Choices From	Abandonment
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

Table 4–60 describes the parameters for the choice group for the advisor Get Abandonment Propensity.

Table 4–60 Choice Group for Advisor Get Abandonment Propensity

Parameter	Description
Choice Group Name	Abandonment
Choice Attributes	Name Likelihood = Get Choice Likelihood ("AbandonmentModel", this)
Scores	Maximize Likelihood = Likelihood
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 4–61 describes the parameters for the model for the advisor Get Abandonment Propensity.

Table 4–61 Model for Advisor Get Abandonment Propensity

Parameter	Description
Model Name	Abandonment Model (choice model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Abandonment
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None
Model Name	Abandonment Model (choice model)

Dependencies:

- Informant - Close Session
- Function - Get Choice Likelihood (Model Name, Choice)

4.5.2.2 Get Likely Web Action

The advisor Get Likely Web Action predicts the most likely Web Action that a particular customer will perform next.

Table 4–62 describes the parameters for the advisor Get Likely Web Action.

Table 4–62 Advisor Get Likely Web Action

Parameter	Description
Advisor Name	Get Likely Web Action
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	None
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Likely Web Action
Group Decision	Select Likely Web Action
Default Choices	None
Logic	None
Pre-condition	None

Table 4–63 describes the parameters for the decision for the advisor Get Likely Web Action.

Table 4–63 Decision for Advisor Get Likely Web Action

Parameter	Description
Decision Name	Select Likely Web Action
Select Choices From	Web Actions
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

Table 4–64 describes the parameters for the choice group for the advisor Get Likely Web Action.

Table 4–64 Choice Group for Advisor Get Likely Web Action

Parameter	Description
Choice Group Name	Web Actions
Choice Attributes	Name Likelihood = Get Choice Likelihood ("WebActionModel", this)
Scores	Maximize Likelihood = Likelihood
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 4–65 describes the parameters for the model for the advisor Get Likely Web Action.

Table 4–65 Model for Advisor Get Likely Web Action

Parameter	Description
Model Name	Web Action Model (choice model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Web Actions
Mutually Exclusive	No
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Customer Action
Temporary Data Storage	None
Model Name	Web Action Model (choice model)

Dependencies:

- Informant - Customer Action
- Function - Get Choice Likelihood (Model Name, Choice)

4.5.2.3 Get Likely Web Duration

The advisor Get Likely Web Duration predicts the length of time that a particular customer will spend on the site.

Table 4–66 describes the parameters for the advisor Get Likely Web Duration.

Table 4–66 Advisor Get Likely Web Duration

Parameter	Description
Advisor Name	Get Likely Web Duration
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	None
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Likely Web Duration
Group Decision	Select Likely Web Duration
Default Choices	None
Logic	None
Pre-condition	None

Table 4–67 describes the parameters for the decision for the advisor Get Likely Web Duration.

Table 4–67 Decision for Advisor Get Likely Web Duration

Parameter	Description
Decision Name	Select Likely Web Duration
Select Choices From	Web Site Duration
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

Table 4–68 describes the parameters for the choice group for the advisor Get Likely Web Duration.

Table 4–68 Choice Group for Advisor Get Likely Web Duration

Parameter	Description
Choice Group Name	Web Site Duration
Choice Attributes	Name Likelihood = Get Choice Likelihood ("WebSiteDurationModel", this)
Scores	Maximize Likelihood = Likelihood
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 4–69 describes the parameters for the model for the advisor Get Likely Web Duration.

Table 4–69 Model for Advisor Get Likely Web Duration

Parameter	Description
Model Name	Web Site Duration Model (choice model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Web Site Duration
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None
Model Name	Web Site Duration Model (choice model)

Dependencies:

- Informant - Customer Action
- Function - Get Choice Likelihood (Model Name, Choice)

4.5.2.4 Get Web Support Types

The advisor Get Web Support Types returns the likelihood for a customer to accept a chat request or email support if offered.

Table 4–70 describes the parameters for the advisor Get Web Support Types.

Table 4–70 Advisor Get Web Support Types

Parameter	Description
Advisor Name	Get Web Support Types
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	None

Table 4–70 (Cont.) Advisor Get Web Support Types

Parameter	Description
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Web Supports
Group Decision	Select Web Supports
Default Choices	None
Logic	None
Pre-condition	None

Table 4–71 describes the parameters for the decision for the advisor Get Web Support Types.

Table 4–71 Decision for Advisor Get Web Support Types

Parameter	Description
Decision Name	Select Web Support Types
Select Choices From	Web Support Types
Number of Choices to Select	2
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("presented");}

Table 4–72 describes the parameters for the choice group for the advisor Get Web Support Types.

Table 4–72 Choice Group for Advisor Get Web Support Types

Parameter	Description
Choice Group Name	Web Support Types
Choice Attributes	Name Likelihood of Usage = Predicted by Web Support Usage Model: Used Threshold = 0.5
Scores	Maximize Likelihood = Likelihood of Usage
Choice Events	Presented Used
Choice Eligibility	Choice / Likelihood of Usage > choice / Threshold
Group Attributes	None
Group Eligibility	None

Table 4–72 (Cont.) Choice Group for Advisor Get Web Support Types

Parameter	Description
Dynamic Choices	None

Table 4–73 describes the parameters for the model for the advisor Get Web Support Types.

Table 4–73 Model for Advisor Get Web Support Types

Parameter	Description
Model Name	Web Support Usage Model (choice event model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Web Support Types
Base Event	Presented
Positive Events	Used
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

Dependencies:

- Informant - Web Support Feedback

4.6 Functions

This section describes the functions used in the RTD_Base_ECommerce Inline Service.

Table 4–74 RTD_Base_ECommerce Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Ad List	None	Advertisements (Array of Ad)	Used for Advertisements dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Ad entity for Advertisement dynamic choice. The result is either ranked advertisements array supplied as incoming parameter of advisors or array of advertisements loaded using data source mapping.
Get Agent Interaction Reasons In Past Days	Past Agent Interactions (Array of Agent Interaction), Days (Integer)	Interaction Reasons (Array of String)	Used for deriving values for Interaction History entity.	This function returns array of interaction reason in past given days.
Get Agent Interaction Types In Past Days	Past Agent Interactions (Array of Agent Interaction), Days (Integer)	Interaction Types (Array of String)	Used for deriving values for Interaction History entity.	This function returns array of interaction types in past given days.

Table 4–74 (Cont.) RTD_Base_ECommerce Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Campaign Names In Past Days	Campaign Items (Array of Campaign Item), Days (Integer)	Campaign Names (Array of String)	Used for deriving values for Campaign History entity	This function returns array of campaign names delivered in past given days.
Get Choice Likelihood	Model Name (String), Choice (Choice)	Likelihood (Double)	Used for scoring in Abandonment, Web Actions, Web Site Duration choice groups	This function returns likelihood for a given inputted choice that is part of a choice model (as opposed to a choice event model). As inputs, the user must pass in a choice and the model name that choice is part of.
Get Cross Sell Product List	Product Id (String) to be used as base product for cross sell	Cross Sell Products (Array of Product)	Used for Cross Sell dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Product entity for Cross Sell dynamic choice. The result is either ranked products array supplied as incoming parameter of advisors or array of cross sell products loaded using data source mapping.
Get Customer Type	Person (Person), Organization (Organization)	Customer Type (String)	Used by Customer entity.	This function is used to determine customer type.
Get Days Left	Expiration Date (Date)	Number of Days Left (Integer)	Used by Promotion entity.	This function calculates the number of days from now to expiryDate.
Get Days Since Last Agent Interaction	Past Agent Interactions (Array of Agent Interaction)	Number of Days (Integer)	Used by Interaction History entity.	This function calculates the number of days from last agent interaction to now.
Get Days Since Last Campaign	Campaign Items (Array of Campaign Item)	Number of Days (Integer)	Used by Campaign History entity.	This function calculates the number of days from last campaign to now.
Get Days Since Last Purchase	Purchased Items (Array of Purchased Item)	Number of Days (Integer)	Used by Purchase History entity.	This function calculates the number of days from last purchase to now.
Get Days Since Last Web Interaction	Past Web Interactions (Array of Web Interaction)	Number of Days (Integer)	Used by Interaction History entity.	This function calculates the number of days from last web interaction to now.
Get Duration In Days	Start Date (Date), End Date (Date)	Number of Days (Integer)	Used by Promotion entity and various other functions.	This function calculates the number of days from start date to end date.
Get Duration In Minutes	Start Time (Date), End Time (Date)	Number of Minutes (Integer)	NA.	This function calculates the number of minutes from start time to end time.
Get Gender	Male (Boolean)	Gender In Text (String)	Used by Customer entity.	This function determines the gender of a person. This converts the given gender type from boolean to string. In Male case, genderMale is boolean value=true fetched from data source.

Table 4–74 (Cont.) RTD_Base_ECommerce Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Interaction Types In Past Days	Past Agent Interactions (Array of Agent Interaction), Past Web Interactions (Array of Web Interaction), Days (Integer)	Interaction Type in Text (String)	Used by Interaction History entity.	This function returns interaction type in past given days.
Get Last Agent Interaction Status	Past Agent Interactions (Array of Agent Interaction)	Last Status (String)	Used by Interaction History entity	This function returns last agent interaction status.
Get Last Agent Interaction Type	Past Agent Interactions (Array of Agent Interaction)	Last Type (String)	Used by Interaction History entity	This function returns last agent interaction type.
Get Last Campaign Category	Campaign Items (Array of Campaign Item)	Category (String)	Used by Campaign History entity.	This function returns last campaign category.
Get Last Campaign Delivery Method	Campaign Items (Array of Campaign Item)	Delivery Method (String)	Used by Campaign History entity.	This function returns last campaign delivery method.
Get Last Campaign Name	Campaign Items (Array of Campaign Item)	Name (String)	Used by Campaign History entity.	This function returns last campaign name.
Get Last Campaign Type	Campaign Items (Array of Campaign Item)	Type (String)	Used by Campaign History entity.	This function returns last campaign type.
Get Last Interaction Type	Days Since Last Agent Interaction (Integer), Days Since Last Web Interaction (Integer)	Interaction Type (String)	Used by Interaction History entity.	This function returns last interaction type. Return value will be Agent, Web, or Both.
Get Last Purchase Amount	Purchased Items (Array of Purchased Item)	Purchase Amount (Double)	Used by Purchase History entity.	This function returns last purchase amount.
Get Last Purchased Product	Purchased Items (Array of Purchased Item)	Product Name (String)	Used by Purchase History entity.	This function returns last purchased product name.
Get Last Purchased Product Line	Purchased Items (Array of Purchased Item)	Product Line (String)	Used by Purchase History entity.	This function returns last purchased product line.
Get Number Of Agent Interaction In Past Days	Past Agent Interactions (Array of Agent Interaction), Days (Integer)	Count (Integer)	Used by Interaction History entity.	This function counts number of agent interactions in past given days.

Table 4–74 (Cont.) RTD_Base_ECommerce Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Number Of Web Interaction In Past Days	Past Web Interactions (Array of Web Interaction), Days (Integer)	Count (Integer)	Used by Interaction History entity.	This function counts number of web interactions in past given days.
Get Performed Web Action Names In Past Days	Past Web Interactions (Array of Web Interaction), Days (Integer)	Action Names (Array of String)	Used by Interaction History entity.	This function returns array of web action names performed in past given days.
Get Product Lines Owned	Purchased Items (Array of Purchased Item)	Product Lines (Array of String)	Used by Purchase History entity.	This function returns product lines of customer owned.
Get Promotion List	None	Promotions (Array of Promotion)	Used for Promotion dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Promotion entity for Promotion dynamic choice. The result is either ranked promotion array supplied as incoming parameter of advisors or array of promotions loaded using data source mapping.
Get Specific Choice Likelihood	Model Name (String), Choice Name (String)	Likelihood (Double)	None. See similar function: Get Choice Likelihood	This function returns likelihood for a given inputted choice that is part of a choice model (as opposed to a choice event model). As inputs, the user must pass in the value for the Name attribute assigned to the choice and the model name that choice is part of.
Get Total Amount Spent	Purchased Items (Array of Purchased Item)	Amount (Double)	Used by Purchase History entity.	This function sums up the amount of customer spent so far.
Get Total Amount Spent in Last 90 Days	Purchased Items (Array of Purchased Item)	Amount (Double)	Used by Purchase History entity.	This function sums up the amount of customer spent in past 90 days.
Get Total Duration In Minutes	Start Time (Date)	Number of Minutes (Integer)	Used by Web Interaction entity.	This function calculates number of minutes between the given start time and now.
Get Up Sell Product List	Product Id (String) to be used as base product for up sell	Up Sell Products (Array of Product)	Used for Up Sell dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Product entity for Up Sell dynamic choice. The result is either ranked products array supplied as incoming parameter of advisors or array of up sell products loaded using data source mapping.
Get Year	Date (Date)	Year (Integer)	Used by Customer entity.	This function returns the year part of Date.
Is Web Support Type Eligible	Web Support Types Choice (Web Support Types Choice)	Eligible (Boolean)	Used by Web Support Types choice group.	This function returns the eligibility of a Web Support Types Choice.
Maximum	X (Integer), Y (Integer)	Maximum (Integer)	Used by Interaction History entity.	This function returns the greater parameter.
Minutes To Now	Start Time (Date)	Minutes (Integer)	Used by Total Duration In Minutes function	This function returns time between given time in the past and now.
Multiply	a (Double, b (Double)	Result (Double)	Used by various RTD elements	This function multiplies the given parameters.
Property Reflect	NA	NA	NA	This function is used for testing. This dumps the values of session attributes.

Table 4–74 (Cont.) RTD_Base_ECommerce Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Select Product Id	None	Product Id (String)	Used as base product for Up Sell and Cross Sell dynamic choices.	This function determines what Product Id to be used as base Product Id for Up Sell or Cross Sell.
Set Choice Event Model	Choice Event Model Name (String), Choice Name (String), Choice Event (String)	None	Used by Web Support Feedback informant.	This function is used to store all of the logic needed to set the choice models tied to the choice groups.
Set Choice Model	Choice Model Name (String), Choice Name (String)	None	Used by Customer Action and Close Session informants.	This function is used to store all of the logic needed to set the choice models tied to the choice groups.
Set Session Rank Offers	Offers (Array of String)	None	Used by all offer-oriented advisors to set the optionally supplied string array of keys into session for later use as dynamic choice source. See corresponding functions: Get Ad List, Get Promotion List, Get Cross Sell Product List, Get Up Sell Product List.	This procedure is used to parse and set Rank Offers before decision selection.
Years To Now	Start Time (Date)	Years (Integer)	Used by Customer entity.	This function calculates the number of years from given date to now.

Base Customer Service

This chapter describes the elements in the Inline Service RTD_Base_Customer_Service. It contains the following topics:

5.1 Introduction to Base Customer Service

This chapter describes the configuration of the Inline Service, RTD_Base_Customer_Service. This Inline Service is independent of any specific front-end application or back-end data schema. As a standalone solution, users will adapt this solution by integrating it into their current environment, and selecting the components of the Inline Service that apply to their own workflow.

This Base Inline Service contains an extensive entity model that can be used to model customer profile attributes as well as current interaction data related to their service interactions. Combining data about the customer and what the customer is doing allows Oracle RTD to perform real time analysis of the interaction and provide predictions on "next best actions" within the context of the current set of actions. These predictions are based on a combination of user-defined rules and likelihood scores determined by Oracle RTD's modeling.

The next best actions include predicting the best offer or promotion to present to the customer, or in some cases basic messaging, where an offer presentation is deemed to be inappropriate, for example, in escalation or complaint calls.

RTD_Base_Customer_Service contains a variety of integration points to allow users to incorporate Oracle RTD directly into their workflow. As a result of letting Oracle RTD learn on current interaction data passed to it through these integration points, Oracle RTD can model not only the results of next best actions, for example, offers, promotions, messaging, but also the interaction itself. This allows users of this Inline Service to model and ultimately predict interaction based attributes such as Interaction Duration, Interaction Reason, and Customer Attrition. As a result, customers can better understand the driving factors behind the metrics that matter to them.

The workflow described in this chapter applies to a generic customer service interaction and supports the following scenarios:

- **Real-time analysis of interaction behavior and customer response to offers and actions**

  By integrating directly with a customer driven application, Oracle RTD can be fed a variety of interaction attributes to develop real-time predictive models. Using Decision Center, users can then navigate through the many reports to help identify drivers of areas such as interaction duration, customer attrition, or offer response.

  RTD_Base_Customer_Service contains the preconfigured choice groups to model these attributes against both the current interaction data and the customer profile data that it has access to. All of the modeling done by Oracle RTD can then be used for future predictions as in the cases mentioned in the following scenarios.

- **Proactive customer assessment, for example, predicting and analyzing in real time, Interaction Reason, Interaction Duration, and Escalation**

  By learning on attributes of an interaction as they occur, Oracle RTD can model them against both current interaction data as well as customer profile data. Using these models, Oracle RTD has the capability to proactively assess a customer to create an interaction profile.

  This information can then be used ahead of time to determine how best to handle a customer. Knowing this information allows agents, for example, to better prepare and handle live interactions with customers prior to taking a call.

- **Providing next best actions or offers in the context of inbound customer interactions**

  Customers can interact with businesses for many reasons. A natural consequence of the variety of these interactions is the complexity of determining what the next best action should be when dealing with a customer.

  At times, it may make sense to suggest a cross-sell or upsell offer if the interaction is about a currently owned product.

  There are other times, however, as in interactions resulting in a complaint, where presenting on offer may not be the best use of the agent's and customer's time. Instead, in situations like this, perhaps a promotional one-time discount or a recommended escalation to a VIP service queue may be a better option.

  With Oracle RTD as a central decisioning engine, users of Oracle RTD have the capability to arbitrate between the catalog of actions and offers a business may have to determine which is most likely to have a favorable response. With RTD_Base_Customer_Service, as current interaction data is fed to the platform, these decisions are made using not only customer profile data, but also real time data about the interaction to help optimize the decision.

- **Identifying and ranking solutions for customer interactions**

  As a general decisioning platform, Oracle RTD can rank and predict a variety of different choice types. For RTD_Base_Customer_Service, configuration is included where Oracle RTD can help rank multiple solution offerings to determine the best solution to present to a customer. As with offers and actions, ranked solutions can be determined by using either Oracle RTD's models, or a customer's own priorities through the use of scoring and eligibility rules.

5.2 Business Process Flows

This section describes a general service workflow with the integration points supported by RTD_Base_Customer_Service, followed by an example of a Contact Center interaction flow that has been enhanced by RTD_Base_Customer_Service.

This section consists of the following topics:

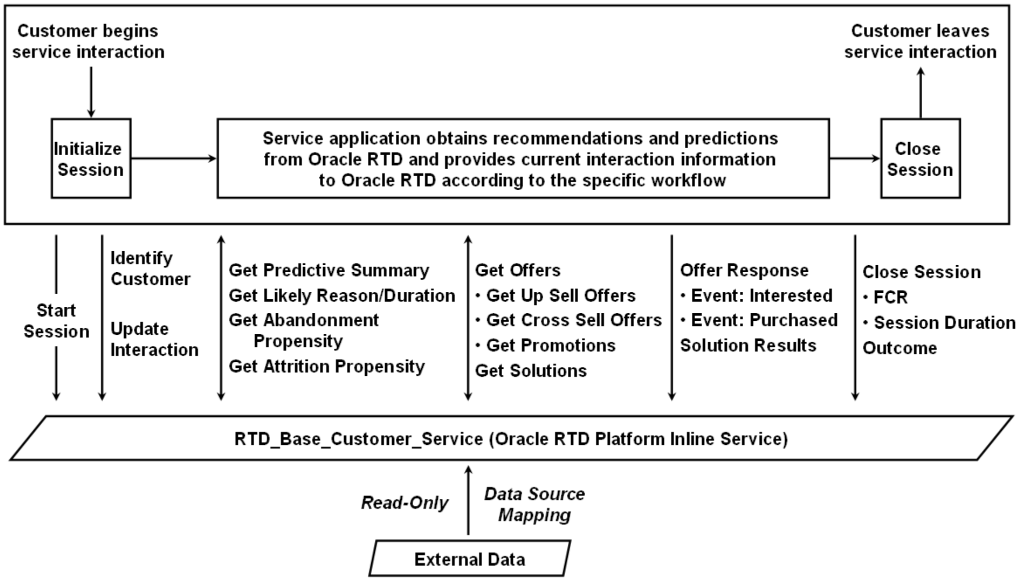
- Section 5.2.1, "General Service Workflow"
- Section 5.2.2, "Service Interaction Workflow"

5.2.1 General Service Workflow

The Inline Service RTD_Base_Customer_Service contains a variety of Oracle RTD Informant and Advisor integration points that users can implement into their workflow process. As depicted in the diagram that follows, these integration points can be arranged in any fashion to support the workflow where Oracle RTD is to be integrated to. Not all integration points need to be utilized for the Inline Service to function correctly. Users can configure and customize each integration point according to their own specific requirements.

The following diagram shows an overview of the stages of a typical service application and several of its possible interactions with the RTD_Base_Customer_Service Inline Service.

Note: The description in the long box in the middle of the workflow is intentionally generic, to represent a general customer application. Exact details will be provided by the customer's own specific interaction workflow.



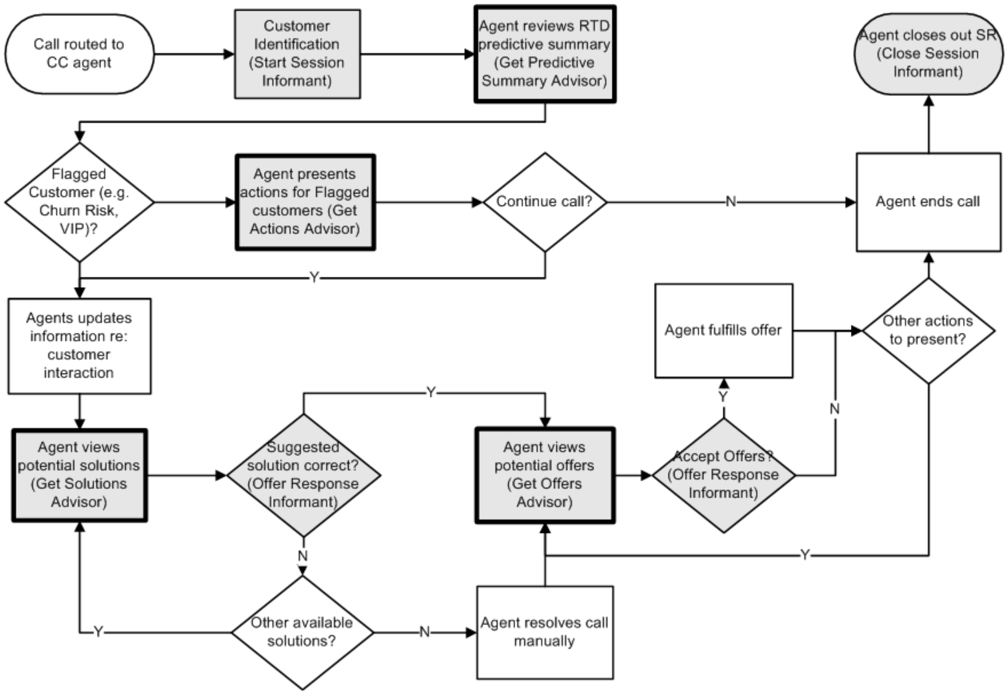
5.2.2 Service Interaction Workflow

The flow chart that follows depicts a sample service interaction flow that has been enabled by the Oracle RTD Inline Service RTD_Base_Customer_Service.

Shaded process boxes indicate touch points for Oracle RTD Integration Points, as follows:

- Shaded boxes with a standard thickness border indicate touch points for Informants
- Shaded boxes with a thicker, bold border indicate touch points for Advisors

These touch points indicate which integration points from the base solution have been integrated. The diagram is as an example of how a user can design the integration points to use and sequence them in the order that makes the most sense.


Call routed to CC agent
Customer Identification (Start Session Informant)
Agent reviews RTD predictive summary (Get Predictive Summary Advisor)
Agent closes out SR (Close Session Informant)
Flagged Customer (e.g. Churn Risk, VIP)?
Agent presents actions for Flagged customers (Get Actions Advisor)
Continue call?
N
Agent ends call
Y
Agents updates information re: customer interaction
Agent fulfills offer
Other actions to present?
Y
Y
N
Agent views potential solutions (Get Solutions Advisor)
Suggested solution correct? (Offer Response Informant)
Agent views potential offers (Get Offers Advisor)
Accept Offers? (Offer Response Informant)
Y
N
Other available solutions?
Y
N
Agent resolves call manually

In this example, the user has chosen to use the Inline Service RTD_Base_Customer_Service to provide multiple predictions for the interaction workflow. The Oracle RTD Advisors and Informants enable the user application to achieve the following objectives:

1. The user aims to have Oracle RTD provide a predictive profile of the customer to alert the service agent of the most likely interaction profile (such as interaction reason) and if the customer is a flagged customer.
2. Depending on the customer's "flagged" status, the intention is to allow Oracle RTD to suggest the best likely action for this customer and have the agent determine if the call should move forward, For example, if the customer has an outstanding bill, Oracle RTD can alert the agent to reroute the call to a billing specialist.
3. Assuming that the call continues, based on the customer profile and what the customer is calling about, Oracle RTD can rank a list of suggested solutions to handle the customers needs.
4. After the customer's main interaction reason is addressed, as a final step, Oracle RTD can then be used to suggest any possible offers or promotions that the agent can tell the customer about.

In order for Oracle RTD to provide the best predictions for each customer interaction, it is critical that the Inline Service is configured to have access to both historical and profile data about the customer as well as the current interaction data.

Information from both the front-end application as well as existing data sources allows Oracle RTD to build models that ultimately predict unique scores for each customer and for each individual interaction. For example, if Oracle RTD was not informed of the real time interaction, similar offers could be recommended for complaint calls as well as support request calls.

Therefore, it is important for users to consider how they would pass data to the RTD server for each interaction (Informants) and what data can be made readily available from existing sources (Data Sources).

Considering the service interaction workflow that appears in this section, the user has chosen to implement the following operations:

1. To call an Oracle RTD informant as soon as the customer is identified, to allow Oracle RTD to begin to build out its session data with back-end historical and profile data about the customer.
2. As more information is gathered during the interaction, to call Oracle RTD to update the session with current interaction data, for example, Interaction Reason, Interaction Product.
3. As each solution is tried, to alert Oracle RTD of which solutions were tried and which ultimately solved the customer's issue.
4. For any offers presented to the customer, to alert Oracle RTD of which offers were presented and if any were accepted by the customer.
5. After the interaction has been completed, to provide Oracle RTD with the final outcomes of the interaction and to allow Oracle RTD to update existing models with the new data.

Ultimately, by applying from this Inline Service the components that are most useful for their workflow, users will be able to reduce their application design and configuration time.

5.3 Performance Goals

The performance goals included with this Inline Service are:

- Maximize Expected Revenue
- Maximize Likelihood
- Maximize Marketing Priority
- Minimize Handle Time
- Minimize Operating Costs

Each performance goal is scored according to the scoring parameters for the specific choice group that uses them. In some cases, not all of the performance goals are used at the same time for any given decision. See Section 5.4, "Choice Groups" for further details. The scoring methods tied to each performance goal can be completely reconfigured to support customer requirements.

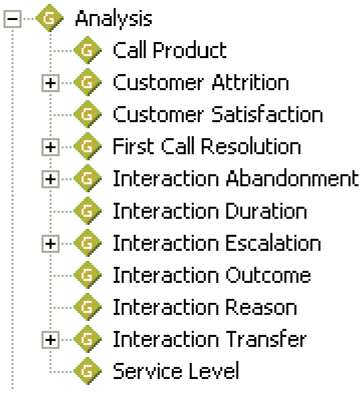
5.4 Choice Groups

This section consists of the following topics:

- Section 5.4.1, "Analysis-Oriented Choice Groups"
- Section 5.4.2, "Decision-Oriented Choice Groups"

5.4.1 Analysis-Oriented Choice Groups

For the business flows described in Section 5.2, "Business Process Flows," the following analysis-oriented choice groups have been configured in this Inline Service. Most of these choice groups are tied to dynamic choices, while choice groups with "Yes/No" values as choices are tied to static choices.



This section contains descriptions of the following choice groups:

- Call Product Choice Group
- Customer Attrition Choice Group
- Customer Satisfaction Choice Group
- First Call Resolution Choice Group
- Interaction Abandonment Choice Group
- Interaction Duration Choice Group
- Interaction Escalation Choice Group
- Interaction Outcome Choice Group
- Interaction Reason Choice Group
- Interaction Transfer Choice Group
- Service Level Choice Group

5.4.1.1 Call Product Choice Group

This choice group contains choices associated with Call Product. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–1 describes the configuration parameters for the Call Product choice group, including a list of the choice attributes.

Table 5–2 describes the choice attributes in more detail.

Table 5–3 describes the dynamic choice parameters for this choice group.

Table 5–1 Configuration Parameters for Call Product Choice Group

Parameter	Description
Choice Group Name	Call Product

Table 5–1 (Cont.) Configuration Parameters for Call Product Choice Group

Parameter	Description
Choice Attributes	Dynamic Choice Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–2.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	Product List ▪ Product List Entity is populated by the Get Product List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–3.

Table 5–2 Choice Attribute Details for Call Product Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Analysis Value	Used to store the singular entity value pulled from the Call Product entity array group attribute.
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs CallProductModel and "this" choice.
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–3 Dynamic Choice Details for Call Product Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Product List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.1.2 Customer Attrition Choice Group

This choice group contains choices associated with Customer Attrition. Its choices are static and configured in the Inline Service. The configured choices are:

- Attrition

- No Attrition

Table 5–4 describes the configuration parameters for the Customer Attrition choice group, including a list of the choice attributes.

Table 5–5 describes the choice attributes in more detail.

Table 5–4 Configuration Parameters for Customer Attrition Choice Group

Parameter	Description
Choice Group Name	Customer Attrition
Choice Attributes	Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–5.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 5–5 Choice Attribute Details for Customer Attrition Choice Group

Choice Attribute	Type	Value
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs CustomerAttritionModel and "this" choice.
Name	String	Specified at the choice level.
Type	String	Specified at the choice level.

5.4.1.3 Customer Satisfaction Choice Group

This choice group contains choices associated with Customer Satisfaction. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–6 describes the configuration parameters for the Customer Satisfaction choice group, including a list of the choice attributes.

Table 5–7 describes the choice attributes in more detail.

Table 5–8 describes the dynamic choice parameters for this choice group.

Table 5–6 Configuration Parameters for Customer Satisfaction Choice Group

Parameter	Description
Choice Group Name	Customer Satisfaction

Table 5–6 (Cont.) Configuration Parameters for Customer Satisfaction Choice Group

Parameter	Description
Choice Attributes	Dynamic Choice Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–7.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	Satisfaction Values List ▪ Customer Satisfaction List Entity is populated by the Get Analysis Values List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–8.

Table 5–7 Choice Attribute Details for Customer Satisfaction Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Analysis Value	Used to store the singular entity value pulled from the Satisfaction Values entity array group attribute.
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs CustomerSatisfactionModel and "this" choice.
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–8 Dynamic Choice Details for Customer Satisfaction Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Satisfaction Values List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.1.4 First Call Resolution Choice Group

This choice group contains choices associated with First Call Resolution. Its choices are static and configured in the Inline Service. The configured choices are:

- FCR Not Resolved
- FCR Resolved

Table 5–9 describes the configuration parameters for the First Call Resolution choice group, including a list of the choice attributes.

Table 5–10 describes the choice attributes in more detail.

Table 5–9 Configuration Parameters for First Call Resolution Choice Group

Parameter	Description
Choice Group Name	First Call Resolution
Choice Attributes	Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–10.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 5–10 Choice Attribute Details for First Call Resolution Choice Group

Choice Attribute	Type	Value
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs FirstCallResolutionModel and "this" choice.
Name	String	Specified at the choice level.
Type	String	

5.4.1.5 Interaction Abandonment Choice Group

This choice group contains choices associated with Interaction Abandonment. Its choices are static and configured in the Inline Service. The configured choices are:

- Abandoned
- Not Abandoned

Table 5–11 describes the configuration parameters for the Interaction Abandonment choice group, including a list of the choice attributes.

Table 5–12 describes the choice attributes in more detail.

Table 5–11 Configuration Parameters for Interaction Abandonment Choice Group

Parameter	Description
Choice Group Name	Interaction Abandonment
Choice Attributes	Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–12.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 5–12 Choice Attribute Details for Interaction Abandonment Choice Group

Choice Attribute	Type	Value
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs InteractionAbandonmentModel and "this" choice.
Name	String	Specified at the choice level.
Type	String	Specified at the choice level.

5.4.1.6 Interaction Duration Choice Group

This choice group contains choices associated with Interaction Duration. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–13 describes the configuration parameters for the Interaction Duration choice group, including a list of the choice attributes.

Table 5–14 describes the choice attributes in more detail.

Table 5–15 describes the dynamic choice parameters for this choice group.

Table 5–13 Configuration Parameters for Interaction Duration Choice Group

Parameter	Description
Choice Group Name	Interaction Duration

Table 5–13 (Cont.) Configuration Parameters for Interaction Duration Choice Group

Parameter	Description
Choice Attributes	Dynamic Choice Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–14.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	Interaction Duration List ■ Interaction Duration List Entity is populated by the Get Analysis Values List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–15.

Table 5–14 Choice Attribute Details for Interaction Duration Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Analysis Value	Used to store the singular entity value pulled from the Interaction Duration entity array group attribute.
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs InteractionDurationModel and "this" choice.
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–15 Dynamic Choice Details for Interaction Duration Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Interaction Duration List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.1.7 Interaction Escalation Choice Group

This choice group contains choices associated with Interaction Escalation. Its choices are static and configured in the Inline Service. The configured choices are:

- Escalated

- Not Escalated

Table 5–16 describes the configuration parameters for the Interaction Escalation choice group, including a list of the choice attributes.

Table 5–17 describes the choice attributes in more detail.

Table 5–16 Configuration Parameters for Interaction Escalation Choice Group

Parameter	Description
Choice Group Name	Interaction Escalation
Choice Attributes	Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–17.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 5–17 Choice Attribute Details for Interaction Escalation Choice Group

Choice Attribute	Type	Value
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs InteractionEscalationModel and "this" choice.
Name	String	
Type	String	

5.4.1.8 Interaction Outcome Choice Group

This choice group contains choices associated with Interaction Outcome. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–18 describes the configuration parameters for the Interaction Outcome choice group, including a list of the choice attributes.

Table 5–19 describes the choice attributes in more detail.

Table 5–20 describes the dynamic choice parameters for this choice group.

Table 5–18 Configuration Parameters for Interaction Outcome Choice Group

Parameter	Description
Choice Group Name	Interaction Outcome
Choice Attributes	Dynamic Choice Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–19.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	Interaction Outcome List ▪ Interaction Outcome List Entity is populated by the Get Analysis Values List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–20.

Table 5–19 Choice Attribute Details for Interaction Outcome Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Analysis Value	Used to store the singular entity value pulled from the Interaction Outcome entity array group attribute.
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs InteractionOutcomeModel and "this" choice.
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–20 Dynamic Choice Details for Interaction Outcome Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Interaction Outcome List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.1.9 Interaction Reason Choice Group

This choice group contains choices associated with Interaction Reason. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–21 describes the configuration parameters for the Interaction Reason choice group, including a list of the choice attributes.

Table 5–22 describes the choice attributes in more detail.

Table 5–23 describes the dynamic choice parameters for this choice group.

Table 5–21 Configuration Parameters for Interaction Reason Choice Group

Parameter	Description
Choice Group Name	Interaction Reason
Choice Attributes	Dynamic Choice Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–22.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	Interaction Reason List ■ Interaction Reason List Entity is populated by the Get Analysis Values List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–23.

Table 5–22 Choice Attribute Details for Interaction Reason Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Analysis Value	Used to store the singular entity value pulled from the Interaction Reason List entity array group attribute.
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs InteractionReason ListModel and "this" choice.
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–23 Dynamic Choice Details for Interaction Reason Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Interaction Reason List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.1.10 Interaction Transfer Choice Group

This choice group contains choices associated with Interaction Transfer. Its choices are static and configured in the Inline Service. The configured choices are:

- Transferred
- Not Transferred

Table 5–24 describes the configuration parameters for the Interaction Transfer choice group, including a list of the choice attributes.

Table 5–25 describes the choice attributes in more detail.

Table 5–24 Configuration Parameters for Interaction Transfer Choice Group

Parameter	Description
Choice Group Name	Interaction Transfer
Choice Attributes	Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–25.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 5–25 Choice Attribute Details for Interaction Transfer Choice Group

Choice Attribute	Type	Value
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs InteractionTransferModel and "this" choice.
Name	String	Specified at the choice level.
Type	String	Specified at the choice level.

5.4.1.11 Service Level Choice Group

This choice group contains choices associated with Service Level of the Interaction. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–26 describes the configuration parameters for the Service Level choice group, including a list of the choice attributes.

Table 5–27 describes the choice attributes in more detail.

Table 5–28 describes the dynamic choice parameters for this choice group.

Table 5–26 Configuration Parameters for Service Level Choice Group

Parameter	Description
Choice Group Name	Service Level
Choice Attributes	Dynamic Choice Category Choice Id Choice Likelihood Name Type For choice attribute details, see Table 5–27.
Scores	For the performance goal Maximize Likelihood, score is the choice attribute Choice Likelihood.
Choice Events	None
Choice Eligibility	None
Group Attributes	Service Level List ■ Service Level List Entity is populated by the Get Analysis Values List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–28.

Table 5–27 Choice Attribute Details for Service Level Choice Group

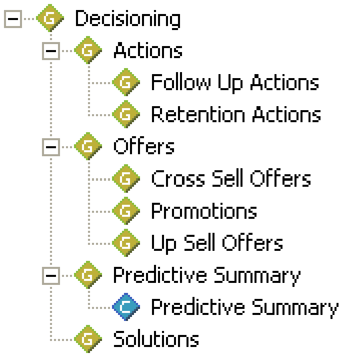
Choice Attribute	Type	Value
Dynamic Choice	Analysis Value	Used to store the singular entity value pulled from the Service Level entity array group attribute.
Category	String	None
Choice Id	String	None
Choice Likelihood	Double	Calculated by the function Get Choice Likelihood, with the inputs ServiceLevelModel and "this" choice.
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–28 Dynamic Choice Details for Service Level Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Service Level List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.2 Decision-Oriented Choice Groups

For the business flows described in Section 5.2, "Business Process Flows," the following decision-oriented choice groups have been configured in this Inline Service. All of these choice groups are tied to dynamic choices and a corresponding choice event model. See Section 5.8, "Models" for further details.



- Decisioning
  - Actions
    - Follow Up Actions
    - Retention Actions
  - Offers
    - Cross Sell Offers
    - Promotions
    - Up Sell Offers
  - Predictive Summary
    - Predictive Summary
  - Solutions

This section contains descriptions of the following choice groups:

- Follow Up Actions Choice Group
- Retention Actions Choice Group
- Cross Sell Offers Choice Group
- Promotions Choice Group
- Up Sell Offers Choice Group
- Predictive Summary Choice Group
- Solutions Choice Group

5.4.2.1 Follow Up Actions Choice Group

This choice group contains choices associated with Follow Up Actions. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–29 describes the configuration parameters for the Follow Up Actions choice group, including a list of the choice attributes.

Table 5–30 describes the choice attributes in more detail.

Table 5–31 describes the dynamic choice parameters for this choice group.

Table 5–29 Configuration Parameters for Follow Up Actions Choice Group

Parameter	Description
Choice Group Name	Follow Up Actions
Choice Attributes	Dynamic Choice Action Id Category Description Text External Rule Marketing Priority Name Type For choice attribute details, see Table 5–30.
Scores	For the performance goal Maximize Expected Revenue, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Likelihood, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Marketing Priority, score is the choice attribute Marketing Priority.
Choice Events	Accepted Interested
Choice Eligibility	None
Group Attributes	Follow Up Action List ■ Follow Up Action List Entity is populated by the Get Follow Up Action List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–31.

Table 5–30 Choice Attribute Details for Follow Up Actions Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Action	Used to store the singular entity value pulled from the Follow Up Action entity array group attribute.
Action Id	String	Dynamic Choice / Action Id
Category	String	None
Description Text	String	Dynamic Choice / Description Text
External Rules	String	None
Marketing Priority	Double	Dynamic Choice / Marketing Priority
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–31 Dynamic Choice Details for Follow Up Actions Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Follow Up Action List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.2.2 Retention Actions Choice Group

This choice group contains choices associated with Retention Actions. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–32 describes the configuration parameters for the Retention Actions choice group, including a list of the choice attributes.

Table 5–33 describes the choice attributes in more detail.

Table 5–34 describes the dynamic choice parameters for this choice group.

Table 5–32 Configuration Parameters for Retention Actions Choice Group

Parameter	Description
Choice Group Name	Retention Actions
Choice Attributes	Dynamic Choice Action Id Category Description Text External Rule Marketing Priority Name Type For choice attribute details, see Table 5–33.
Scores	For the performance goal Maximize Expected Revenue, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Likelihood, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Marketing Priority, score is the choice attribute Marketing Priority.
Choice Events	Accepted Interested
Choice Eligibility	None
Group Attributes	Retention Action List ▪ Retention Action List Entity is populated by the Get Retention Action List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None

Table 5–32 (Cont.) Configuration Parameters for Retention Actions Choice Group

Parameter	Description
Dynamic Choices	For details, see Table 5–34.

Table 5–33 Choice Attribute Details for Retention Actions Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Action	Used to store the singular entity value pulled from the Retention Action entity array group attribute.
Action Id	String	Dynamic Choice / Action Id
Category	String	None
Description Text	String	Dynamic Choice / Description Text
External Rules	String	None
Marketing Priority	Double	Dynamic Choice / Marketing Priority
Name	String	Dynamic Choice / Name
Type	String	Dynamic Choice / Type

Table 5–34 Dynamic Choice Details for Retention Actions Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Retention Action List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.2.3 Cross Sell Offers Choice Group

This choice group contains choices associated with Cross Sell Offers. Its choices are configured dynamically and are determined by users based on their customer workflow. This choice group relies on either a provided product id (see Group attributes) in order to determine which dynamic choices should be populated dynamically according to the interaction product, or a list of cross sell offers provided as an attribute to the Get Offers or Get Cross Sell offers integration point.

Table 5–35 describes the configuration parameters for the Cross Sell Offers choice group, including a list of the choice attributes.

Table 5–36 describes the choice attributes in more detail.

Table 5–37 describes the dynamic choice parameters for this choice group.

Table 5–35 Configuration Parameters for Cross Sell Offers Choice Group

Parameter	Description
Choice Group Name	Cross Sell Offers

Table 5–35 (Cont.) Configuration Parameters for Cross Sell Offers Choice Group

Parameter	Description
Choice Attributes	Dynamic Choice Category Cost External Rule Marketing Priority Name Offer ID Offer Type Revenue For choice attribute details, see Table 5–36.
Scores	For the performance goal Maximize Expected Revenue, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Likelihood, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Marketing Priority, score is the choice attribute Marketing Priority.
Choice Events	Fulfilled Interested Presented
Choice Eligibility	None
Group Attributes	Cross Sell Products ▪ Cross Sell Product List Entity populated by the Get Cross Sell Product List function whose input is the interaction product id. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–37.

Table 5–36 Choice Attribute Details for Cross Sell Offers Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Action	Used to store the singular entity value pulled from the Cross Sell Products entity array group attribute.
Category	String	Dynamic Choice / Category
Cost	Double	Dynamic Choice / Unit Price
External Rules	String	None
Marketing Priority	Double	Dynamic Choice / Marketing Priority
Name	String	Dynamic Choice / Name
Offer Id	String	Dynamic Choice / Product Id
Offer Type	String	"Cross Sell"
Revenue	Double	None

Table 5–37 Dynamic Choice Details for Cross Sell Offers Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Cross Sell Products
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Offer Id
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.2.4 Promotions Choice Group

This choice group contains choices associated with Promotions. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–38 describes the configuration parameters for the Promotions choice group, including a list of the choice attributes.

Table 5–39 describes the choice attributes in more detail.

Table 5–40 describes the dynamic choice parameters for this choice group.

Table 5–38 Configuration Parameters for Promotions Choice Group

Parameter	Description
Choice Group Name	Promotions
Choice Attributes	Dynamic Choice Category Cost External Rule Marketing Priority Name Offer ID Offer Type Revenue For choice attribute details, see Table 5–39.
Scores	For the performance goal Maximize Expected Revenue, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Likelihood, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Marketing Priority, score is the choice attribute Marketing Priority.
Choice Events	Fulfilled Interested Presented
Choice Eligibility	None
Group Attributes	Promotion List ■ Promotion List Entity populated by the Get Promotion List function. See Section 5.5, "Entities" for more details.

Table 5–38 (Cont.) Configuration Parameters for Promotions Choice Group

Parameter	Description
Group Eligibility	None
Dynamic Choices	For details, see Table 5–40.

Table 5–39 Choice Attribute Details for Promotions Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Action	Used to store the singular entity value pulled from the Promotions entity array group attribute.
Category	String	Dynamic Choice / Category
Cost	Double	None
External Rules	String	None
Marketing Priority	Double	Dynamic Choice / Marketing Priority
Name	String	Dynamic Choice / Name
Offer Id	String	Dynamic Choice / Promotion Id
Offer Type	String	"Promotions"
Revenue	Double	None

Table 5–40 Dynamic Choice Details for Promotions Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Promotion List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Offer Id
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.2.5 Up Sell Offers Choice Group

This choice group contains choices associated with Up Sell Offers. Its choices are configured dynamically and are determined by users based on their customer workflow. This choice group relies on either a provided product id (see Group attributes) in order to determine which dynamic choices should be populated dynamically according to the interaction product, or a list of up sell offers provided as an attribute to the Get Offers or Get Up Sell offers integration point.

Table 5–41 describes the configuration parameters for the Up Sell Offers choice group, including a list of the choice attributes.

Table 5–42 describes the choice attributes in more detail.

Table 5–43 describes the dynamic choice parameters for this choice group.

Table 5–41 Configuration Parameters for Up Sell Offers Choice Group

Parameter	Description
Choice Group Name	Up Sell Offers

Table 5–41 (Cont.) Configuration Parameters for Up Sell Offers Choice Group

Parameter	Description
Choice Attributes	Dynamic Choice Category Cost External Rule Marketing Priority Name Offer ID Offer Type Revenue For choice attribute details, see Table 5–42.
Scores	For the performance goal Maximize Expected Revenue, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Likelihood, score is determined by likelihood from corresponding choice event model. For the performance goal Maximize Marketing Priority, score is the choice attribute Marketing Priority.
Choice Events	Fulfilled Interested Presented
Choice Eligibility	None
Group Attributes	Up Sell Products ■ Up Sell Product List Entity populated by the Get Up Sell Product List function whose input is the interaction product id. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–43.

Table 5–42 Choice Attribute Details for Up Sell Offers Choice Group

Choice Attribute	Type	Value
Dynamic Choice	Action	Used to store the singular entity value pulled from the Up Sell Products entity array group attribute.
Category	String	Dynamic Choice / Category
Cost	Double	Dynamic Choice / Unit Price
External Rules	String	None
Marketing Priority	Double	Dynamic Choice / Marketing Priority
Name	String	Dynamic Choice / Name
Offer Id	String	Dynamic Choice / Product Id
Offer Type	String	"Up Sell"
Revenue	Double	None

Table 5–43 Dynamic Choice Details for Up Sell Offers Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Up Sell Products
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Offer Id
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.4.2.6 Predictive Summary Choice Group

The Predictive Summary choice group contains the single static choice, **Predictive Summary**. It is used in conjunction with the **Select Predictive Summary** decision and returns a collective summary of call detail decisions related to the customer.

Table 5–44 describes the configuration parameters for the Predictive Summary choice group, including a list of the choice attributes.

Table 5–45 describes the choice attributes in more detail.

Table 5–44 Configuration Parameters for Predictive Summary Choice Group

Parameter	Description
Choice Group Name	Predictive Summary
Choice Attributes	Abandonment Likelihood Attrition Likelihood Escalation Likelihood First Call Resolution Likelihood Interaction Duration Likelihood Interaction Reason Likelihood Likely Interaction Duration Likely Interaction Reason Likely Product Product Likelihood Transfer Likelihood For choice attribute details, see Table 5–45.
Scores	None
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 5–45 Choice Attribute Details for Predictive Summary Choice Group

Choice Attribute	Type	Value
Abandonment Likelihood	Double	Calculated by the Get Specific Choice Likelihood function for the parameters InteractionAbandonmentModel and Abandoned.

Table 5–45 (Cont.) Choice Attribute Details for Predictive Summary Choice Group

Choice Attribute	Type	Value
Attrition Likelihood	Double	Calculated by the Get Specific Choice Likelihood function for the parameters CustomerAttritionModel and Attrition.
Escalation Likelihood	Double	Calculated by the Get Specific Choice Likelihood function for the parameters InteractionEscalationModel and Escalated.
First Call Resolution Likelihood	Double	Calculated by the Get Specific Choice Likelihood function for the parameters FirstCallResolutionModel and FcrResolved.
Interaction Duration Likelihoods	Double	Calculated by the Get Specific Choice Likelihood function for the parameters InteractionDurationModel and the value returned from the function Get Top Interaction Duration.
Interaction Reason Likelihoods	Double	Calculated by the Get Specific Choice Likelihood function for the parameters InteractionReasonModel and the value returned from the function Get Top Interaction Reason.
Likely Interaction Duration	String	Determined by the function Get Top Interaction Duration.
Likely Interaction Reason	String	Determined by the function Get Top Interaction Reason.
Likely Product	String	Determined by the function Get Top Call Product.
Product Likelihood	Double	Calculated by the Get Specific Choice Likelihood function for the parameters CallProduct and the value returned from the function Get Top Call Product.
Transfer Likelihood	Double	Calculated by the Get Specific Choice Likelihood function for the parameters InteractionTransferModel and Transferred.

5.4.2.7 Solutions Choice Group

This choice group contains choices associated with Solutions. Its choices are configured dynamically and are determined by users based on their customer workflow.

Table 5–46 describes the configuration parameters for the Solutions choice group, including a list of the choice attributes.

Table 5–47 describes the choice attributes in more detail.

Table 5–48 describes the dynamic choice parameters for this choice group.

Table 5–46 Configuration Parameters for Solutions Choice Group

Parameter	Description
Choice Group Name	Solutions
Choice Attributes	Description Text Dynamic Choice Name Product Name Solution ID For choice attribute details, see Table 5–47.
Scores	For the performance goal Maximize Likelihood, score is determined by likelihood from corresponding choice event model.

Table 5–46 (Cont.) Configuration Parameters for Solutions Choice Group

Parameter	Description
Choice Events	Accepted Presented
Choice Eligibility	None
Group Attributes	Solution List ■ Solution List Entity populated by the Get Solution List function. See Section 5.5, "Entities" for more details.
Group Eligibility	None
Dynamic Choices	For details, see Table 5–48.

Table 5–47 Choice Attribute Details for Solutions Choice Group

Choice Attribute	Type	Value
Description Text	String	Dynamic Choice / Description Text
Dynamic Choice	Solution	Used to store the singular entity value pulled from the Solution entity array group attribute.
Name	String	Dynamic Choice / Name
Product Name	String	
Solution Id	String	Dynamic Choice / Solution Id

Table 5–48 Dynamic Choice Details for Solutions Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Solution List
Choice attribute to assign the entity data	Dynamic Choice
Entity attribute that contains the choices id	Name
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

5.5 Entities

There are two types of entity in the Inline Service RTD_Base_Customer_Service, namely session entities and entities used for dynamic choices.

This section consists of the following topics:

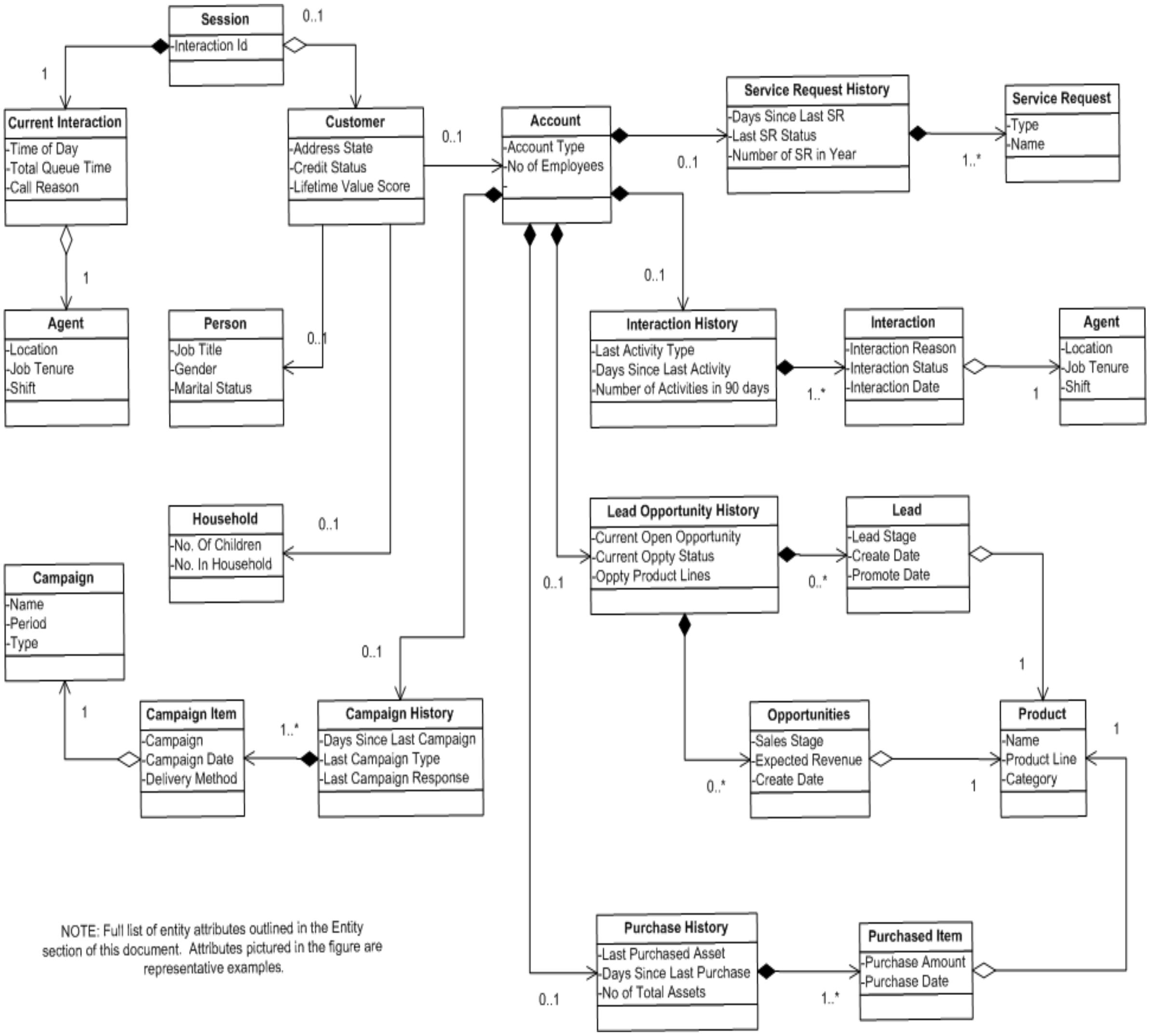
- Section 5.5.1, "Session Entity Model"
- Section 5.5.2, "Session Entities"
- Section 5.5.3, "Dynamic Choice Entity Models"
- Section 5.5.4, "Dynamic Choice Entities"

5.5.1 Session Entity Model

The Inline Service RTD_Base_Customer_Service contains a predefined session entity model that models attributes related to both the profile of the Oracle RTD current interaction and the customer's profile and history. In addition to these learning entities, all of the required entities for mapping a choice group to its dynamic choices are also defined.

As database table mappings are not defined in the base application, users must map their own data to the provided logical model. Each entity included in the Inline Service is fully extensible. This allows users to add or remove attributes that are available to their workflow.

The following diagram shows the entities, together with some of their attributes, and the relationships between the entities defined in RTD_Base_Customer_Service.



NOTE: Full list of entity attributes outlined in the Entity section of this document. Attributes pictured in the figure are representative examples.

Notation

The diagram shows the standard notation used in UML class diagrams, with each directed line representing a relationship *from* element A *to* element B, as follows:

- A line with no diamond represents an **association** between elements A and B.
- A line with a clear diamond represents an **aggregation**, with element A "owning" element B. However, with aggregation, removing an instance of element A does not remove the corresponding B element instances.

- A line with a filled diamond represents a **composition**, with element A "owning" element B. For a composition, removing an instance of element A also removes the corresponding B element instances.

The multiplicity of a relationship restricts how many element B instances the relationship may have. The restriction denotes either a precise limit, such as 1 or 0..1, or an open-ended upper limit, such as "zero or more" or "one or more."

5.5.2 Session Entities

Session entities have a primary use for storing the attributes used for model learning.

This section describes the following entities:

- Session Entity
- Account
- Agent
- Campaign
- Campaign History
- Campaign Item
- Current Interaction
- Customer
- Household
- Interaction
- Interaction History
- Lead
- Lead Opportunity History
- Opportunity
- Person
- Purchased Item
- Purchase History
- Rank Offers
- Service Request
- Service Request History

5.5.2.1 Session Entity

(Key = Interaction Id)

Table 5–49 Session Entity

Attribute	Type	Array	Comments
Current Interaction	Current Interaction	No	None
Customer	Customer	No	None
Interaction Churn Propensity	Double	No	None

Table 5–49 (Cont.) Session Entity

Attribute	Type	Array	Comments
Rank Offers	Rank Offers	No	Utilized when ranking offers directly from an Offer advisor instead of read from a data source.

5.5.2.2 Account

The Account entity is associated with the session through the Customer entity.

Table 5–50 Account Entity

Attribute	Type	Array	Comments
Account Type	String	No	None
Annual Revenue	Double	No	None
Industry	Industry	No	None
Line of Business	String	No	None
No of Employees	Double	No	None
Number Of Years Established	Integer	No	None
Region	String	No	None

5.5.2.3 Agent

The Agent entity contains attributes associated with the agent interacting with the customer.

Table 5–51 Agent Entity

Attribute	Type	Array	Comments
Work Queue	String	No	None
Shift Period	String	No	None
Tenure	Integer	No	None
Training Level	String	No	None

5.5.2.4 Campaign

The Campaign entity is a learning entity that is referenced by the Campaign History entity of the customer. It is initialized by the Identify Customer informant.

Table 5–52 Campaign Entity

Attribute	Type	Array	Comments
Campaign Id	String	No	Not used for analysis
Category	String	No	None
Name	String	No	None
Period	String	No	None
Type	String	No	None

5.5.2.5 Campaign History

The Campaign History entity is a learning entity that contains campaign history of the customer. It is initialized by the Identify Customer informant.

Table 5–53 Campaign History Entity

Attribute	Type	Array	Comments
Campaign Items	Campaign Item	Yes	None
Campaigns In Past 30 Days	String	Yes	Calculated by the function Get Campaign Names In Past Days
Days Since Last Campaign	Integer	No	Calculated by the function Get Days Since Last Campaign
Last Campaign Category	String	No	Calculated by the function Get Last Campaign Category
Last Campaign Delivery Method	String	No	Calculated by the function Get Last Campaign Delivery Method
Last Campaign Name	String	No	Calculated by the function Get Last Campaign Name
Last Campaign Type	String	No	Calculated by the function Get Last Campaign Type

5.5.2.6 Campaign Item

The Campaign Item entity is a learning entity that is referenced by the Campaign History entity of the customer. It is initialized by the Identify Customer informant.

Table 5–54 Campaign Item Entity

Attribute	Type	Array	Comments
Campaign	Campaign	No	None
Campaign Date	Date	No	None
Delivery Method	String	No	None

5.5.2.7 Current Interaction

The Current Interaction entity is a learning entity that contains all of the attributes tied to the session's current interaction.

Table 5–55 Current Interaction Entity

Attribute	Type	Array	Comments
Abandonment Status	String	No	None
Agent	Agent	No	None
Agent Id	String	No	None
Attrition Status	String	No	None
Call Center Location	String	No	None
Duration	String	No	None
Escalation Status	String	No	None
Final Solution	String	No	None
Hold Time	Integer	No	None

Table 5–55 (Cont.) Current Interaction Entity

Attribute	Type	Array	Comments
Interaction Date	Date	No	None
Interaction Product Id	String	No	None
Interaction Reason	String	No	None
IVR Path	String	No	None
Outcome	String	No	None
Priority	String	No	None
Product Name	String	No	None
Resolution	String	No	None
Satisfaction Rating	String	No	None
Service Level	String	No	None
Solutions Offered	String	Yes	None
Status	String	No	None
Transfer Status	String	No	None

5.5.2.8 Customer

The Customer entity is a learning entity that contains all attributes tied to the customer identified in the session.

Table 5–56 Customer Entity

Attribute	Type	Array	Comments
Customer Id	String	No	Not used for analysis
Account	Account	No	None
Account Id	String	No	Not used for analysis
Address City	String	No	None
Address Country	String	No	None
Address Postal Code	String	No	None
Address Region	String	No	None
Address State Province	String	No	None
Campaign History	Campaign History	No	None
Credit Hold	String	No	None
Household	Household	No	None
Household Id	String	No	Not used for analysis
Interaction History	Interaction History	No	None
Lead Opportunity History	Lead Opportunity History	No	None
Lifetime Value Score	Double	No	None
Offline Churn Propensity	Double	No	None

Table 5–56 (Cont.) Customer Entity

Attribute	Type	Array	Comments
Person	Person	No	None
Person Id	String	No	Not used for analysis
Phone Area Code	String	No	None
Purchase History	Purchase History	No	None
Service Request History	Service Request History	No	None
Status	String	No	None
Target Market Segment	String	No	None
Tenure	Double	Yes	None
Total Credit Limit	Double	No	None
Type	String	No	None

5.5.2.9 Household

The Household entity is a learning entity that contains all household related attributes.

Table 5–57 Household Entity

Attribute	Type	Array	Comments
No. in household	Integer	No	None
No. of children	Integer	No	None
No. of Pets	Integer	No	None

5.5.2.10 Interaction

The Interaction entity is a learning entity that contains interaction history details of the customer. It is initialized by the Identify Customer informant.

Table 5–58 Interaction Entity

Attribute	Type	Array	Comments
Interaction Id	String	No	Key
Abandonment Status	String	No	None
Agent	Agent	No	None
Agent Id	String	No	Not used for analysis
Escalation Status	String	No	None
Final Solution	String	No	None
Hold Time	Integer	No	None
Interaction Channel	String	No	None
Interaction Date	Date	No	None
Interaction Duration	Integer	No	None
Interaction Reason	String	No	None
Interaction Status	String	No	None

Table 5–58 (Cont.) Interaction Entity

Attribute	Type	Array	Comments
Interaction Type	String	No	None
Outcome	String	No	None
Product Name	String	No	None
Resolution	String	No	None
Satisfaction Rating	String	No	None
Transfer Status	String	No	None

5.5.2.11 Interaction History

The Interaction History entity is a learning entity that contains interaction history of the customer. It is initialized by the Identify Customer informant.

Table 5–59 Interaction History Entity

Attribute	Type	Array	Comments
Days Since Last Interaction	Integer	No	Calculated by the function Get Days Since Last Interaction
Interaction Types In Past 30 Days	String	Yes	Calculated by the function Get Interaction Types In Past Days
Last Interaction Status	String	No	Calculated by the function Get Last Interaction Status
Last Interaction Type	String	No	Calculated by the function Get Last Interaction Type
Number Of Interactions In Past 30 Days	Integer	No	Calculated by the function Get Number Of Interactions In Past Days
Number Of Interactions In Past 90 Days	Integer	No	Calculated by the function Get Number Of Interactions In Past Days
Number Of Interactions In Past Year	Integer	No	Calculated by the function Get Number Of Interactions In Past Days
Past Interactions	Interaction	Yes	Not used for analysis

5.5.2.12 Lead

The Lead entity is associated with the session through the Lead Opportunity History entity.

Table 5–60 Lead Entity

Attribute	Type	Array	Comments
Create Date	Date	No	None
Lead Product	Product	No	None
Opportunity Promotion Date	Date	No	None
Sales Stage	String	No	None

5.5.2.13 Lead Opportunity History

The Lead Opportunity History entity is a learning entity that contains lead opportunity history of the customer. It is initialized by the Identify Customer informant.

Table 5–61 Lead Opportunity History Entity

Attribute	Type	Array	Comments
Days Since Last Opportunity	Integer	No	Calculated by the function Get Days Since Last Opportunity
Last Opportunity Expected Revenue	String	No	Calculated by the function Get Last Opportunity Expected Revenue
Last Opportunity Product	String	No	Calculated by the function Get Last Opportunity Product
Last Opportunity Product Line	String	No	Calculated by the function Get Last Opportunity Product Line
Last Opportunity Sales Stage	Integer	No	Calculated by the function Get Last Opportunity Sales Stage
Number Of Opportunities In Past Year	Integer	No	Calculated by the function Get Number Of Opportunities In Past Days
Opportunity Product Lines	String	Yes	Calculated by the function Get Opportunity Product Lines
Past Leads	Lead	Yes	None
Past Opportunities	Opportunity	Yes	None

5.5.2.14 Opportunity

The Opportunity entity is associated with the session through the Lead Opportunity History entity.

Table 5–62 Opportunity Entity

Attribute	Type	Array	Comments
Create Date	Date	No	None
Expected Revenue	Double	No	None
Opportunity Product	Product	No	None
Sales Stage	String	No	None

5.5.2.15 Person

The Person entity is associated with the session through the Customer entity.

Table 5–63 Person Entity

Attribute	Type	Array	Comments
Age	Integer	No	None
Annual Income	Double	No	None
Ethnicity	String	No	None
Gender	String	No	None
Job Title	String	No	None

Table 5–63 (Cont.) Person Entity

Attribute	Type	Array	Comments
Marital Status	String	No	None
Preferred Language	String	No	None

5.5.2.16 Purchased Item

The Purchased Item entity contains purchase instance data of a product used in the customer Purchase History entity.

Table 5–64 Purchased Item Entity

Attribute	Type	Array	Comments
Purchase Amount	Double	No	None
Purchase Date	Date	No	None
Purchased Product	Product	No	None

5.5.2.17 Purchase History

The Purchase History entity is a learning entity that contains purchase history of the customer. It is initialized by the Identify Customer informant.

Table 5–65 Purchase History Entity

Attribute	Type	Array	Comments
Average Purchase Amount	Double	No	None
Days Since Last Purchase	Integer	No	Calculated by the function Get Days Since Last Purchase
Last Purchased Amount	Double	No	Calculated by the function Get Last Last Purchased Amount
Last Purchased Product	String	No	Calculated by the function Get Last Purchased Product
Last Purchased Product Line	String	No	Calculated by the function Get Last Purchased Product Line
Product Lines Owned	String	Yes	Calculated by the function Get Product Lines Owned
Purchased Items	Purchased Item	Yes	None
Total Amount Spent	Double	No	Calculated by the function Get Total Amount Spent
Total Amount Spent In Last 90 Days	Double	No	Calculated by the function Get Total Amount Spent In Last 90 Days

5.5.2.18 Rank Offers

The Rank Offers entity is a transient attribute for holding an array of offer id's to be ranked supplied as incoming parameter of Get offer advisor calls.

Table 5–66 Rank Offers Entity

Attribute	Type	Array	Comments
CrossSellOffers	String	Yes	None

Table 5–66 (Cont.) Rank Offers Entity

Attribute	Type	Array	Comments
In Use	Boolean	No	None
Promotions	String	Yes	None
UpSellOffers	String	Yes	None

5.5.2.19 Service Request

The Service Request entity is a singular entity used in the Service Request History customer entity.

Table 5–67 Service Request Entity

Attribute	Type	Array	Comments
Account Id	String	No	Key
Agent	Agent	No	None
Agent Id	String	No	Not used for analysis
Closed Date	Date	No	None
Create Date	Date	No	None
Product	String	No	None
Product Id	String	No	None
Reason	String	No	None
Solution Id	String	No	None
Solution Name	String	No	None
Status	String	No	None
Type	String	No	None

5.5.2.20 Service Request History

The Service Request History entity is a learning entity that contains the Service Request history of the customer. It is initialized by the Identify Customer informant.

Table 5–68 Service Request History Entity

Attribute	Type	Array	Comments
Current Open SR	Boolean		None
Days Since Last Service Request	Integer	No	Calculated by the function Get Days Since Last Service Request
Last Service Request Product	String	No	Calculated by the function Get Last Service Request Product
Last Service Request Reason	String	No	Calculated by the function Get Last Service Request Reason
Last Opportunity Sales Solution	Integer	No	Calculated by the function Get Last Service Request Solution
Last Service Request Status	String	No	Calculated by the function Get Last Service Request Status
Last Opportunity Sales Type	Integer	No	Calculated by the function Get Last Service Request Type

Table 5–68 (Cont.) Service Request History Entity

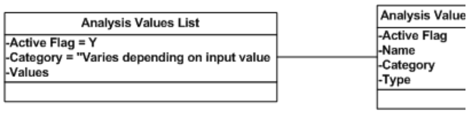
Attribute	Type	Array	Comments
Past Service Requests	Service Request	Yes	Not used for analysis
Service Requests In Past 90 Days	Integer	No	Calculated by the function Get Number Of Service Requests In Past Days
Service Requests In Past Year	Integer	No	Calculated by the function Get Number Of Service Requests In Past Days

5.5.3 Dynamic Choice Entity Models

In addition to the entities created for session learning, Inline Service entities used for dynamic choices are also defined for this solution. These entities are used to populate the specific choices for each of the choice groups configured to used dynamic choices. Further details are available in Section 5.4, "Choice Groups."

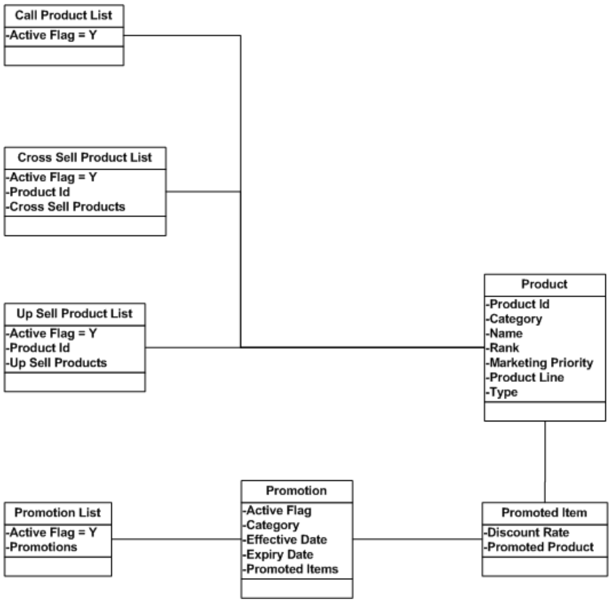
5.5.3.1 Analysis Value

The following diagram depicts the relations of the List entities specified with the Analysis Value entity:



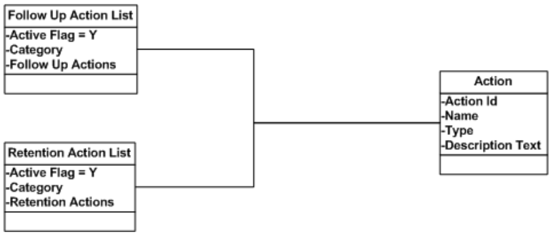
5.5.3.2 Product

The following diagram depicts the relations of the List entities specified with the Product entity:



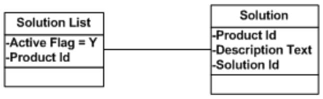
5.5.3.3 Action

The following diagram depicts the relations of the List entities specified with the Action entity:



5.5.3.4 Solution

The following diagram depicts the relations of the List entities specified with the Solution entity:



5.5.4 Dynamic Choice Entities

As a general note, each dynamic choice group will have a corresponding set of entities - a "list" entity and an "individual" entity. For example, the Promotions dynamic choice group is associated with a Promotion entity and a Promotions List entity. The attributes of these entities are also the choice attributes for promotion.

This section describes the following entities:

- Action
- Analysis Value
- Analysis Values List
- Cross Sell Product List
- Follow Up Action List
- Product
- Product List
- Promoted Item
- Promotion
- Promotion List
- Retention Action List
- Solution
- Solution List
- Up Sell Product List

5.5.4.1 Action

The Action entity is the singular entity used in the dynamic list entities Follow Up Action List and Retention Action List.

Table 5–69 Action Entity

Attribute	Type	Array	Comments
Action Id	String	No	None

Table 5–69 (Cont.) Action Entity

Attribute	Type	Array	Comments
Description Text	String	No	None
Marketing Priority	Double	No	None
Name	String	No	None
Type	String	No	None

5.5.4.2 Analysis Value

The Analysis Value entity is the singular entity used in all of the Analysis List based entities including Interaction Duration List, Interaction Outcome List, and Interaction Reason List. The List entities are used for populating the dynamic choices for the choice groups of the same name.

Table 5–70 Analysis Value Entity

Attribute	Type	Array	Comments
Name	String	No	None
Type	String	No	None

5.5.4.3 Analysis Values List

The Analysis Value List entity is the List entity used to populate the dynamic choices for the Customer Satisfaction, Interaction Reason, Interaction Duration, Interaction Outcome, and Service Level choice groups. The choice group populated by this list is determined by the value for the key attribute, Category, for example, Category = Interaction Duration, and so on.

Table 5–71 Analysis Values List Entity

Attribute	Type	Array	Comments
Active Flag	String	No	Key
Category	String	No	Key
Values	Analysis Value	Yes	None

5.5.4.4 Cross Sell Product List

The Cross Sell Product List entity is a collection of cross sell products. It is used for data retrieval to populate the Cross Sell Dynamic Choice Group.

Table 5–72 Cross Sell Product List Entity

Attribute	Type	Array	Comments
Product Id	String	No	None
Cross Sell Products	Product	Yes	None

5.5.4.5 Follow Up Action List

The Follow Up Action List entity is a List entity used to populate dynamic choices for the Follow Up Action choice group.

Table 5–73 Follow Up Action List Entity

Attribute	Type	Array	Comments
Active Flag	String	No	Key
Category	String	No	Key
Follow Up Actions	Action	Yes	None

5.5.4.6 Product

The Product entity is used for product descriptions for the Up Sell Offer, Cross Sell Offer, Purchase History, Leads, and Opportunity entities.

Table 5–74 Product Entity

Attribute	Type	Array	Comments
Product Id	String	No	Key
Category	String	No	None
Marketing Priority	Double	No	None
Name	String	No	None
Popularity Rank	Integer	No	None
Product Line	String	No	None
Type	String	No	None
Unit Price	Double	No	None

5.5.4.7 Product List

The Product List entity is a collection of products. It is used for data retrieval to populate the Call Product Choice Group.

Table 5–75 Product List Entity

Attribute	Type	Array	Comments
Active Flag	String	No	Key
Products	Product	Yes	None

5.5.4.8 Promoted Item

The Promoted Item entity contains an instance of a promoted product used in the Promotion entity.

Table 5–76 Promoted Item Entity

Attribute	Type	Array	Comments
Discount Rate	Double	No	None
Promoted Product	Product	Yes	None

5.5.4.9 Promotion

The Promotion entity is a singular entity of a Promotion used in the Promotion List entity.

Table 5–77 Promotion Entity

Attribute	Type	Array	Comments
Promotion Id	String	No	None
Category	String	No	None
Days Left	Integer	No	Calculated by the function Get Days Left
Duration In Days	Integer	No	Calculated by the function Get Duration In Days
Effective Date	Date	No	None
Expiry Date	Date	No	None
Marketing Priority	Double	No	None
Name	String	No	None
Period	String	No	None
Promoted Items	Promoted Item	Yes	None
Type	String	No	None

5.5.4.10 Promotion List

The Promotion List entity is a collection of Promotions. It is used for data retrieval to populate the Promotions Dynamic Choice Group.

Table 5–78 Promotion List Entity

Attribute	Type	Array	Comments
Active	String	No	Key
Promotions	Promotion	Yes	None

5.5.4.11 Retention Action List

The Retention Action List entity is a List entity used to populate dynamic choices for the Retention Action choice group.

Table 5–79 Retention Action List Entity

Attribute	Type	Array	Comments
Active Flag	String	No	Key
Category	String	No	Key
Retention Actions	Action	Yes	None

5.5.4.12 Solution

The Solution entity is a singular entity used in the Solution List entity.

Table 5–80 Solution Entity

Attribute	Type	Array	Comments
Description Text	String	No	None
Name	String	No	None
Product Name	String	No	None

Table 5–80 (Cont.) Solution Entity

Attribute	Type	Array	Comments
Solution Id	String	No	None

5.5.4.13 Solution List

The Solution List entity is a List entity used to populate dynamic choices for the Solutions choice group.

Table 5–81 Solution List Entity

Attribute	Type	Array	Comments
Active Flag	String	No	Key
Product Id	String	No	Key
Solutions	Solution	No	None

5.5.4.14 Up Sell Product List

The Up Sell Product List entity is a collection of Up Sell products. It is used for data retrieval to populate the Up Sell Dynamic Choice Group.

Table 5–82 Up Sell Product List Entity

Attribute	Type	Array	Comments
Product Id	String	No	None
Up Sell Products	Product	Yes	None

5.6 Informants

The following informants are configured in the Inline Service RTD_Base_Customer_Service:

- Action Response
- Close Session
- Identify Customer
- Offer Response
- Solution Result
- Start Session
- Update Customer Interaction

5.6.1 Action Response

The Action Response informant can be called to record the results of actions recommended by Oracle RTD.

Table 5–83 describes the parameters for the Action Response informant.

Table 5–83 Informant Action Response

Parameter	Description
Informant Name	Action Response

Table 5–83 (Cont.) Informant Action Response

Parameter	Description
Session Keys	Session / Interaction Id
Request Data	Action Id (String) Action Result (String) Action Type (String)
External System	CRM System
Order	20
Force session close	No
Logic	Logic is written to record the appropriate Event for the given action id.
Pre-condition	None

5.6.2 Close Session

The Close Session informant records the final outcomes of the customer interaction and updates all analytical choice models at session close.

Table 5–84 describes the parameters for the Close Session informant.

Table 5–84 Informant Close Session

Parameter	Description
Informant Name	Close Session
Session Keys	Session / Interaction Id
Request Data	Abandonment Status (String) mapped to Current Interaction.Abandoment Status Attrition Status (String) mapped to Current Interaction.Attrition Status Customer Satisfaction (String) mapped to Current Interaction.Satisfaction Rating Escalation Status (String) mapped to Current Interaction.Escalation Status FCR Resolved (String) mapped to Current Interaction.FCR Resolved Interaction Duration (String) mapped to Current Interaction.Duration Interaction Outcome (String) mapped to Current Interaction.Outcome Resolution (String) mapped to Current Interaction.Resolution Solutions Offered (String Array) mapped to Current Interaction.Solutions Offered Transfer Status (String) mapped to Current Interaction.Transfer Status
External System	CRM System
Order	25
Force session close	Yes
Logic	Logic is written to update all choice models based on incoming request data values for this integration point.

Table 5–84 (Cont.) Informant Close Session

Parameter	Description
Pre-condition	None

5.6.3 Identify Customer

The Identify Customer informant sends the id values for Account, Person, and Household to Oracle RTD.

Table 5–85 describes the parameters for the Identify Customer informant.

Table 5–85 Informant Identify Customer

Parameter	Description
Informant Name	Identify Customer
Session Keys	Session / Interaction Id
Request Data	Account Id (String) mapped to Customer.Account Id Household Id (String) mapped to Customer.Household Id Person Id (String) mapped to Customer.Person Id
External System	CRM System
Order	5
Force session close	No
Logic	Logic is written to fill the session().customer() entity.
Pre-condition	None

5.6.4 Offer Response

The Offer Response informant can be called to record the results of offers recommended by Oracle RTD.

Table 5–86 describes the parameters for the Offer Response informant.

Table 5–86 Informant Offer Response

Parameter	Description
Informant Name	Offer Response
Session Keys	Session / Interaction Id
Request Data	Offer Id (String) Event (String) Offer Type (String)
External System	CRM System
Order	70
Force session close	No
Logic	Logic is written to record the appropriate Event for the given offer id.
Pre-condition	None

5.6.5 Solution Result

The Solution Result informant can be called to record the results of solutions offered to the customer.

Table 5–87 describes the parameters for the Solution Result informant.

Table 5–87 Informant Solution Result

Parameter	Description
Informant Name	Solution Result
Session Keys	Session / Interaction Id
Request Data	Solution Id (String Array) Solution Result (String Array)
External System	CRM System
Order	15
Force session close	No
Logic	Logic is written to record the appropriate results for the given array of solution ids.
Pre-condition	None

5.6.6 Start Session

The Start Session informant is called to initiate the Oracle RTD session and sends initial information regarding the customer interaction.

Table 5–88 describes the parameters for the Start Session informant.

Table 5–88 Informant Start Session

Parameter	Description
Informant Name	Start Session
Session Keys	Session / Interaction Id
Request Data	Agent Id (String) mapped to Current Interaction.Agent Id Customer Id (String) mapped to Customer.Customer Id Hold Type (String) mapped to Current Interaction.Hold Time IVR Path (String) mapped to Current Interaction.IVR Path
External System	CRM System
Order	1
Force session close	No
Logic	None
Pre-condition	None

5.6.7 Update Customer Interaction

The Update Customer Interaction informant is called to update attributes regarding the current interaction.

Table 5–89 describes the parameters for the Update Customer Interaction informant.

Table 5–89 Informant Update Customer Interaction

Parameter	Description
Informant Name	Update Customer Interaction
Session Keys	Session / Interaction Id
Request Data	Interaction Product Id (String) mapped to Current Interaction.Interaction Product Id Interaction Reason (String) mapped to Current Interaction.Interaction Reason Priority (String) mapped to Current Interaction.Priority Service Level (String) mapped to Current Interaction.Service Level Status (String) mapped to Current Interaction.Status
External System	CRM System
Order	20
Force session close	No
Logic	None
Pre-condition	None

5.7 Advisors and Associated Decisions

For each advisor listed in this section, a detailed breakdown is provided for the advisor, followed by the decision called by the advisor.

This section contains descriptions of the following advisors (and their corresponding decisions):

- Get Actions
- Get Cross Sell Offers
- Get Follow Up Actions
- Get Likely Call Product
- Get Likely Interaction Duration
- Get Likely Interaction Reason
- Get Offers
- Get Predictive Summary
- Get Promotions
- Get Ranked Solutions
- Get Retention Actions
- Get Up Sell Offers

5.7.1 Get Actions

The advisor Get Actions returns ranked Actions pulled from the choice group of the same name. It utilizes the Select Actions decision.

Table 5–90 describes the parameters for the advisor Get Actions.

Table 5–90 Advisor Get Actions

Parameter	Description
Advisor Name	Get Actions
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System
Order	11
Force session close	No
Decision	Select Actions
Group Decision	Select Actions
Default Choices	None
Logic	None
Pre-condition	None

Table 5–91 describes the parameters for the decision for the advisor Get Actions.

Table 5–91 Decision for Advisor Get Actions

Parameter	Description
Decision Name	Select Actions
Select Choices From	Actions Choice Group
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 33% Maximize Expected Revenue 33% Maximize Marketing Priority 33%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.7.2 Get Cross Sell Offers

This advisor returns ranked Cross Sell Offers to the front end application. It utilizes the Select Cross Sell Offers decision.

Table 5–92 describes the parameters for the advisor Get Cross Sell Offers.

Table 5–92 Advisor Get Cross Sell Offers

Parameter	Description
Advisor Name	Get Cross Sell Offers
Session Keys	Session / Interaction Id

Table 5–92 (Cont.) Advisor Get Cross Sell Offers

Parameter	Description
Request Data	Number of Offers (Integer) - Optional for overriding the default number of offers returned by the advisor. Product Id (String) - Id is to be used as base product for determining which cross sell offers to rank. If not supplied through the Get Cross Sell Offers integration point, this value must be populated through the Update Customer Interaction integration point. Rank Offers (String Array) - Optional to let Oracle RTD rank the supplied array of offer ids instead of retrieving the offers from an external data source.
External System	CRM System
Order	60
Force session close	No
Decision	Select Cross Sell Offers
Group Decision	Select Cross Sell Offers
Default Choices	None
Logic	Logic is included to populate the Rank Offers session entity with the incoming offer ids from the Rank Offers attribute if supplied. After this is done, the Select Cross Sell Offers decision is executed to rank either the supplied offer ids or read from the mapped dynamic choice data source.
Pre-condition	None

Table 5–93 describes the parameters for the decision for the advisor Get Cross Sell Offers.

Table 5–93 Decision for Advisor Get Cross Sell Offers

Parameter	Description
Decision Name	Select Cross Sell Offers
Select Choices From	Cross Sell Offers Choice Group
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 50% Maximize Expected Revenue 50%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.7.3 Get Follow Up Actions

The advisor Get Follow Up Actions returns ranked Follow Up Actions pulled from the choice group of the same name. It utilizes the Select Follow Up Actions decision.

Table 5–94 describes the parameters for the advisor Get Follow Up Actions.

Table 5–94 Advisor Get Follow Up Actions

Parameter	Description
Advisor Name	Get Follow Up Actions
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System
Order	11
Force session close	No
Decision	Select Follow Up Actions
Group Decision	Select Follow Up Actions
Default Choices	None
Logic	None
Pre-condition	None

Table 5–95 describes the parameters for the decision for the advisor Get Follow Up Actions.

Table 5–95 Decision for Advisor Get Follow Up Actions

Parameter	Description
Decision Name	Select Follow Up Actions
Select Choices From	Follow Up Actions Choice Group
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 33% Maximize Expected Revenue 33% Maximize Marketing Priority 33%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.7.4 Get Likely Call Product

The advisor Get Likely Call Product returns ranked call products pulled from the choice group of the same name. It utilizes the Select Likely Call Product decision. This decision is also called in the function, Get Top Call Product, which is used to populate the corresponding attribute in the Predictive Summary choice.

Table 5–96 describes the parameters for the advisor Get Likely Call Product.

Table 5–96 Advisor Get Likely Call Product

Parameter	Description
Advisor Name	Get Likely Call Product

Table 5–96 (Cont.) Advisor Get Likely Call Product

Parameter	Description
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System
Order	7
Force session close	No
Decision	Select Likely Call Product
Group Decision	Select Likely Call Product
Default Choices	None
Logic	None
Pre-condition	None

Table 5–97 describes the parameters for the decision for the advisor Get Likely Call Product.

Table 5–97 Decision for Advisor Get Likely Call Product

Parameter	Description
Decision Name	Select Likely Call Product
Select Choices From	Call Product Choice Group
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

5.7.5 Get Likely Interaction Duration

The advisor Get Likely Interaction Duration returns ranked duration values pulled from the choice group of the same name. It utilizes the Select Likely Interaction Duration decision. This decision is also called in the function, Get Top Interaction Duration, which is used to populate the corresponding attribute in the Predictive Summary choice.

Table 5–98 describes the parameters for the advisor Get Likely Interaction Duration.

Table 5–98 Advisor Get Likely Interaction Duration

Parameter	Description
Advisor Name	Get Likely Interaction Duration
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System

Table 5–98 (Cont.) Advisor Get Likely Interaction Duration

Parameter	Description
Order	7
Force session close	No
Decision	Select Likely Interaction Duration
Group Decision	Select Likely Interaction Duration
Default Choices	None
Logic	None
Pre-condition	None

Table 5–99 describes the parameters for the decision for the advisor Get Likely Interaction Duration.

Table 5–99 Decision for Advisor Get Likely Interaction Duration

Parameter	Description
Decision Name	Select Likely Interaction Duration
Select Choices From	Interaction Duration Choice Group
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

5.7.6 Get Likely Interaction Reason

The advisor Get Likely Interaction Reason returns ranked duration values pulled from the choice group of the same name. It utilizes the Select Likely Interaction Reason decision. This decision is also called in the function, Get Top Interaction Reason, which is used to populate the corresponding attribute in the Predictive Summary choice.

Table 5–100 describes the parameters for the advisor Get Likely Interaction Reason.

Table 5–100 Advisor Get Likely Interaction Reason

Parameter	Description
Advisor Name	Get Likely Interaction Reason
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System
Order	7
Force session close	No
Decision	Select Likely Interaction Reason
Group Decision	Select Likely Interaction Reason

Table 5–100 (Cont.) Advisor Get Likely Interaction Reason

Parameter	Description
Default Choices	None
Logic	None
Pre-condition	None

Table 5–101 describes the parameters for the decision for the advisor Get Likely Interaction Reason.

Table 5–101 Decision for Advisor Get Likely Interaction Reason

Parameter	Description
Decision Name	Select Likely Interaction Reason
Select Choices From	Interaction Reason Choice Group
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

5.7.7 Get Offers

This advisor returns ranked offers to the front end application.

Table 5–102 describes the parameters for the advisor Get Offers.

Table 5–102 Advisor Get Offers

Parameter	Description
Advisor Name	Get Offers
Session Keys	Session / Interaction Id
Request Data	Number of Offers (Integer) - Optional for overriding the default number of offers returned by the advisor. Product Id (String) - Id is to be used as base product for determining which offers to rank. If not supplied through the Get Offers integration point, this value must be populated through the Update Customer Interaction integration point. Rank Offers (String Array) - Optional to let Oracle RTD rank the supplied array of offer ids instead of retrieving the offers from an external data source
External System	CRM System
Order	60
Force session close	No
Decision	Select Offers
Group Decision	Select Offers
Default Choices	None

Table 5–102 (Cont.) Advisor Get Offers

Parameter	Description
Logic	Logic is included to populate the Rank Offers session entity with the incoming offer ids from the Rank Offers attribute if supplied. After this is done, the Select Offers decision is executed to rank either the supplied offer ids or read from the mapped dynamic choice data source.
Pre-condition	None

Table 5–103 describes the parameters for the decision for the advisor Get Offers.

Table 5–103 Decision for Advisor Get Offers

Parameter	Description
Decision Name	Select Offers
Select Choices From	Offers Choice Group
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 50% Maximize Expected Likelihood 50%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.7.8 Get Predictive Summary

The advisor Get Predictive Summary returns the choice Predictive Summary from the choice group of the same name. It utilizes the Select Predictive Summary decision. The choice attributes tied to the predictive summary choice aggregate a combination of the top advisor predictions tied to the analytical choice group, for example, Interaction Reason, Interaction Duration, Attrition Likelihood, and so on.

Table 5–104 describes the parameters for the advisor Get Predictive Summary.

Table 5–104 Advisor Get Predictive Summary

Parameter	Description
Advisor Name	Get Predictive Summary
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System
Order	6
Force session close	No
Decision	Select Predictive Summary
Group Decision	Select Predictive Summary
Default Choices	None

Table 5–104 (Cont.) Advisor Get Predictive Summary

Parameter	Description
Logic	None
Pre-condition	None

Table 5–105 describes the parameters for the decision for the advisor Get Predictive Summary.

Table 5–105 Decision for Advisor Get Predictive Summary

Parameter	Description
Decision Name	Select Predictive Summary
Select Choices From	Predictive Summary Choice Group
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

5.7.9 Get Promotions

This advisor returns ranked promotions to the front end application.

Table 5–106 describes the parameters for the advisor Get Promotions.

Table 5–106 Advisor Get Promotions

Parameter	Description
Advisor Name	Get Promotions
Session Keys	Session / Interaction Id
Request Data	Number of Offers (Integer) - Optional for overriding the default number of promotions returned by the advisor. Product Id (String) - Id is to be used as base product for determining which offers to rank. If not supplied through the Get Promotions integration point, this value must be populated through the Update Customer Interaction integration point. Rank Offers (String Array) - Optional to let Oracle RTD rank the supplied array of offer ids instead of retrieving the offers from an external data source
External System	CRM System
Order	60
Force session close	No
Decision	Select Promotions
Group Decision	Select Promotions
Default Choices	None

Table 5–106 (Cont.) Advisor Get Promotions

Parameter	Description
Logic	Logic is included to populate the Rank Offers session entity with the incoming offer ids from the Rank Offers attribute if supplied. After this is done, the Select Promotions decision is executed to rank either the supplied offer ids or read from the mapped dynamic choice data source.
Pre-condition	None

Table 5–107 describes the parameters for the decision for the advisor Get Promotions.

Table 5–107 Decision for Advisor Get Promotions

Parameter	Description
Decision Name	Select Promotions
Select Choices From	Promotions Choice Group
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 50% Maximize Expected Likelihood 50%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.7.10 Get Ranked Solutions

The advisor Get Ranked Solutions returns ranked duration values pulled from the choice group of the same name. It utilizes the Select Ranked Solutions decision.

Table 5–108 describes the parameters for the advisor Get Ranked Solutions.

Table 5–108 Advisor Get Ranked Solutions

Parameter	Description
Advisor Name	Get Ranked Solutions
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System
Order	11
Force session close	No
Decision	Select Ranked Solutions
Group Decision	Select Ranked Solutions
Default Choices	None
Logic	None
Pre-condition	None

Table 5–109 describes the parameters for the decision for the advisor Get Ranked Solutions.

Table 5–109 Decision for Advisor Get Ranked Solutions

Parameter	Description
Decision Name	Select Ranked Solutions
Select Choices From	Solutions Choice Group
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.7.11 Get Retention Actions

The advisor Get Retention Actions returns ranked Retention Actions pulled from the choice group of the same name. It utilizes the Select Retention Actions decision.

Table 5–110 describes the parameters for the advisor Get Retention Actions.

Table 5–110 Advisor Get Retention Actions

Parameter	Description
Advisor Name	Get Retention Actions
Session Keys	Session / Interaction Id
Request Data	None
External System	CRM System
Order	11
Force session close	No
Decision	Select Retention Actions
Group Decision	Select Retention Actions
Default Choices	None
Logic	None
Pre-condition	None

Table 5–111 describes the parameters for the decision for the advisor Get Retention Actions.

Table 5–111 Decision for Advisor Get Retention Actions

Parameter	Description
Decision Name	Select Retention Actions
Select Choices From	Retention Actions Choice Group

Table 5–111 (Cont.) Decision for Advisor Get Retention Actions

Parameter	Description
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 33% Maximize Expected Revenue 33% Maximize Marketing Priority 33%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.7.12 Get Up Sell Offers

This advisor returns ranked Up Sell Offers to the front end application.

Table 5–112 describes the parameters for the advisor Get Up Sell Offers.

Table 5–112 Advisor Get Up Sell Offers

Parameter	Description
Advisor Name	Get Up Sell Offers
Session Keys	Session / Interaction Id
Request Data	Number of Offers (Integer) - Optional for overriding the default number of Up Sell offers returned by the advisor. Product Id (String) - Id is to be used as base product for determining which Up Sell offers to rank. If not supplied through the Get Up Sell Offers integration point, this value must be populated through the Update Customer Interaction integration point. Rank Offers (String Array) - Optional to let Oracle RTD rank the supplied array of offer ids instead of retrieving the offers from an external data source.
External System	CRM System
Order	60
Force session close	No
Decision	Select Up Sell Offers
Group Decision	Select Up Sell Offers
Default Choices	None
Logic	Logic is included to populate the Rank Offers session entity with the incoming offer ids from the Rank Offers attribute if supplied. After this is done, the Select Up Sell Offers decision is executed to rank either the supplied offer ids or read from the mapped dynamic choice data source.
Pre-condition	None

Table 5–113 describes the parameters for the decision for the advisor Get Up Sell Offers.

Table 5–113 Decision for Advisor Get Up Sell Offers

Parameter	Description
Decision Name	Select Up Sell Offers
Select Choices From	Up Sell Offers Choice Group
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 50% Maximize Expected Revenue 50%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

5.8 Models

This section describes each of the models in the Inline Service RTD_Base_Customer_Service and their configuration parameters.

This section contains the following topics:

- Section 5.8.1, "Choice Models"
- Section 5.8.2, "Choice Event Models"

5.8.1 Choice Models

Choice models have been configured for each of the choice groups under the Analysis parent choice group.

This section describes the following choice models:

- Call Product Model
- Customer Attrition Model
- Customer Satisfaction Model
- First Call Resolution Model
- Interaction Abandonment Model
- Interaction Duration Model
- Interaction Escalation Model
- Interaction Outcome Model
- Interaction Reason Model
- Interaction Transfer Model
- Service Level Model

5.8.1.1 Call Product Model

The Call Product model is a choice model associated with the Call Product choice group.

Table 5–114 Call Product Model

Parameter	Description
Model Name	Call Product Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Call Product
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Interaction Product Id Current Interaction - Product Name
Learn Location	On session close
Temporary Data Storage	None

5.8.1.2 Customer Attrition Model

The Customer Attrition model is a choice model associated with the Customer Attrition choice group.

Table 5–115 Customer Attrition Model

Parameter	Description
Model Name	Customer Attrition Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Customer Attrition
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Attrition Status
Learn Location	On session close
Temporary Data Storage	None

5.8.1.3 Customer Satisfaction Model

The Customer Satisfaction model is a choice model associated with the Customer Satisfaction choice group.

Table 5–116 Customer Satisfaction Model

Parameter	Description
Model Name	Customer Satisfaction Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Customer Satisfaction
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Satisfaction Rating
Learn Location	On session close

Table 5–116 (Cont.) Customer Satisfaction Model

Parameter	Description
Temporary Data Storage	None

5.8.1.4 First Call Resolution Model

The First Call Resolution model is a choice model associated with the First Call Resolution choice group.

Table 5–117 First Call Resolution Model

Parameter	Description
Model Name	First Call Resolution Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	First Call Resolution
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.8.1.5 Interaction Abandonment Model

The Interaction Abandonment model is a choice model associated with the Interaction Abandonment choice group.

Table 5–118 Interaction Abandonment Model

Parameter	Description
Model Name	Interaction Abandonment Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Interaction Abandonment
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Abandonment Status
Learn Location	On session close
Temporary Data Storage	None

5.8.1.6 Interaction Duration Model

The Interaction Duration model is a choice model associated with the Interaction Duration choice group.

Table 5–119 Interaction Duration Model

Parameter	Description
Model Name	Interaction Duration Model

Table 5–119 (Cont.) Interaction Duration Model

Parameter	Description
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Interaction Duration
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Duration
Learn Location	On session close
Temporary Data Storage	None

5.8.1.7 Interaction Escalation Model

The Interaction Escalation model is a choice model associated with the Interaction Escalation choice group.

Table 5–120 Interaction Escalation Model

Parameter	Description
Model Name	Interaction Escalation Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Interaction Escalation
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Escalation Status
Learn Location	On session close
Temporary Data Storage	None

5.8.1.8 Interaction Outcome Model

The Interaction Outcome model is a choice model associated with the Interaction Outcome choice group.

Table 5–121 Interaction Outcome Model

Parameter	Description
Model Name	Interaction Outcome Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Interaction Outcome
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Outcome
Learn Location	On session close
Temporary Data Storage	None

5.8.1.9 Interaction Reason Model

The Interaction Reason model is a choice model associated with the Interaction Reason choice group.

Table 5–122 Interaction Reason Model

Parameter	Description
Model Name	Interaction Reason Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Interaction Reason
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Interaction Reason
Learn Location	On session close
Temporary Data Storage	None

5.8.1.10 Interaction Transfer Model

The Interaction Transfer model is a choice model associated with the Interaction Transfer choice group.

Table 5–123 Interaction Transfer Model

Parameter	Description
Model Name	Interaction Transfer Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Interaction Transfer
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	Current Interaction - Transfer Status
Learn Location	On session close
Temporary Data Storage	None

5.8.1.11 Service Level Model

The Service Level model is a choice model associated with the Service Level choice group.

Table 5–124 Service Level Model

Parameter	Description
Model Name	Service Level Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Service Level
Mutually Exclusive	Yes
Partitioning Attributes	None

Table 5–124 (Cont.) Service Level Model

Parameter	Description
Excluded Attributes	Current Interaction - Service Level
Learn Location	On session close
Temporary Data Storage	None

5.8.2 Choice Event Models

Choice Event models have been configured for each of the choice groups under the Decisioning parent choice group.

This section describes the following choice event models:

- Cross Sell Offers Event Model
- Follow Up Action Event Model
- Offers Event Model
- Promotions Event Model
- Retention Action Event Model
- Solutions Event Model
- Up Sell Offers Event Model

5.8.2.1 Cross Sell Offers Event Model

The Cross Sell Offers Event model is a choice model associated with the Cross Sell Offers choice group.

Table 5–125 Cross Sell Offers Event Model

Parameter	Description
Model Name	Cross Sell Offers Event Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Cross Sell Offers
Base Event	Presented
Positive Outcome Events	Interested Fulfilled
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.8.2.2 Follow Up Action Event Model

The Follow Up Action Event model is a choice model associated with the Follow Up Actions choice group.

Table 5–126 Cross Sell Offers Event Model

Parameter	Description
Model Name	Follow Up Action Event Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Follow Up Actions
Base Event	Presented
Positive Outcome Events	Accepted
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.8.2.3 Offers Event Model

The Offers Event model is a choice model associated with the Offers choice group.

Table 5–127 Offers Event Model

Parameter	Description
Model Name	Offers Event Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Offers
Base Event	Presented
Positive Outcome Events	Interested Fulfilled
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.8.2.4 Promotions Event Model

The Promotions Event model is a choice model associated with the Promotions choice group.

Table 5–128 Promotions Event Model

Parameter	Description
Model Name	Promotions Event Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Promotions
Base Event	Presented
Positive Outcome Events	Interested Fulfilled

Table 5–128 (Cont.) Promotions Event Model

Parameter	Description
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.8.2.5 Retention Action Event Model

The Retention Action Event model is a choice model associated with the Retention Actions choice group.

Table 5–129 Retention Action Event Model

Parameter	Description
Model Name	Retention Action Event Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Retention Actions
Base Event	Presented
Positive Outcome Events	Accepted
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.8.2.6 Solutions Event Model

The Solutions Event model is a choice model associated with the Ranked Solutions choice group.

Table 5–130 Solutions Event Model

Parameter	Description
Model Name	Solutions Event Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Ranked Solutions
Base Event	Presented
Positive Outcome Events	Accepted
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.8.2.7 Up Sell Offers Event Model

The Up Sell Offers Event model is a choice model associated with the Up Sell Offers choice group.

Table 5–131 Up Sell Offers Event Model

Parameter	Description
Model Name	Up Sell Offers Event Model
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Up Sell Offers
Base Event	Presented
Positive Outcome Events	Interested Fulfilled
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

5.9 Functions

This section catalogs the list of predefined functions included in the Inline Service RTD_Base_Customer_Service.

Table 5–132 RTD_Base_Customer_Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Analysis Values List	Category (String))	Analysis Value Array	Groups Attributes for Choice Groups: - Customer Satisfaction - Interaction Outcome - Interaction Reason - Interaction Duration - Service Level	This function retrieves an array of Analysis choice values for various Analysis Choice Groups. It is used to populate the dynamic choices for Customer Satisfaction Choice Group, Interaction Outcome Choice Group, Interaction Reason Choice Group, Interaction Duration Choice Group, and Service Level Choice Group based on the value for the Category input.
Get Campaign Names in Past Days	Campaign Items (Campaign Item Array) Days (Integer)	String Array	Campaign History Entity	This function loops through all campaign items associated with the customer and returns and unique list of campaign name values within the given number of days from the current date.
Get Choice Likelihood	modelName (String) thisChoice (Choice)	Double	Choice attribute, Choice Likelihood.	For a given choice model name and choice, this function returns the RTD likelihood score.
Get Cross Sell Product List	Product Id	Product Array	Cross Sell Product Choice Group, Group Attributes	This function is used to return an array of Products for the Cross Sell dynamic choice. The result is either a ranked products array supplied as incoming parameter of an advisor or an array of cross sell products loaded using data source mapping.
Get Days Left	Expiry Date (Date)	Integer	Promotion Entity	This function calculates the number of days from now to expiryDate.

Table 5–132 (Cont.) RTD_Base_Customer_Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Days Since Last Campaign	Campaign Items (Campaign Item Array)	Integer	Campaign History Entity	This function calculates the number of days from last campaign to now based on the Campaign Date.
Get Days Since Last Interaction	Past Interactions (Interaction Array)	Integer	Interaction History Entity	This function calculates the number of days from last interaction to now based on the Interaction Date.
Get Days Since Last Opportunity	Past Opportunities (Opportunity Array)	Integer	Opportunity History Entity	This function calculates the number of days from last customer opportunity to now based on the Opportunity create date.
Get Days Since Last Purchase	Purchased Items (Purchased Item Array)	Integer	Purchase History Entity	This function calculates the number of days from the last purchase to now based on the purchase date.
Get Days Since Last Service Request	Past Service Requests (Service Request Array)	Integer	Service Request History Entity	This function calculates the number of days from last service request to now based on the Service Request create date.
Get Duration In Days	Start Date (Date) End Date (Date)	Integer	Multiple Objects	This function calculates the number of days from start date to end date.
Get Follow Up Action List	Active (String) Category (String))	Action Entity Array	Follow Up Actions Choice Group, Group Attributes	This function retrieves an array of follow up actions. Used with the Follow Up actions dynamic choice group.
Get Interaction Types In Past Days	Past Interactions (Interaction Array) Days (Integer)	String Array	Interaction History Entity	This function returns array of interaction types within the past given days based on the Interaction create date.
Get Last Campaign Category	Campaign Items (Campaign Item Array)	String	Campaign History Entity	This function returns the campaign category of the most recent campaign item based on campaign date.
Get Last Campaign Delivery Method	Campaign Items (Campaign Item Array)	String	Campaign History Entity	This function returns the campaign delivery method of the most recent campaign item based on campaign date.
Get Last Campaign Name	Campaign Items (Campaign Item Array)	String	Campaign History Entity	This function returns the campaign name of the most recent campaign item based on campaign date.
Get Last Campaign Type	Campaign Items (Campaign Item Array)	String	Campaign History Entity	This function returns the campaign type of the most recent campaign item based on campaign date.
Get Last Interaction Status	Past Interactions (Interaction Array)	String	Interaction History Entity	This function returns the status of the last associated interaction to the customer based on the interaction date.
Get Last Interaction Type	Past Interactions (Interaction Array)	String	Interaction History Entity	This function returns the type of the last associated interaction to the customer based on the interaction date.

Table 5–132 (Cont.) RTD_Base_Customer_Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Last Opportunity Expected Revenue	Past Opportunities (Opportunity Array)	Double	Opportunity History Entity	This function returns the expected revenue of the most recent opportunity associated with the customer.
Get Last Opportunity Product	Past Opportunities (Opportunity Array)	String	Opportunity History Entity	This function returns the product of the most recent opportunity associated with the customer.
Get Last Opportunity Product Line	Past Opportunities (Opportunity Array)	String	Opportunity History Entity	This function returns the product line of the most recent opportunity associated with the customer.
Get Last Opportunity Sales Stage	Past Opportunities (Opportunity Array)	String	Opportunity History Entity	This function returns the current sales stage of the most recent opportunity associated with the customer.
Get Last Purchased Amount	Purchase Items (Purchased Item Array)	Double	Purchase History Entity	This function returns the purchase amount of the most recent purchased product associated with the customer.
Get Last Purchased Product	Purchase Items (Purchased Item Array)	String	Purchase History Entity	This function returns the product name of the most recent purchased product associated with the customer.
Get Last Purchased Product Line	Purchase Items (Purchased Item Array)	String	Purchase History Entity	This function returns the product line of the most recent purchased product associated with the customer.
Get Last Service Request Product	Past Service Requests (Service Request Array)	String	Service Request History Entity	This function returns the product associated with the last created Service Request tied to the customer.
Get Last Service Request Reason	Past Service Requests (Service Request Array)	String	Service Request History Entity	This function returns the reason associated with the last created Service Request tied to the customer.
Get Last Service Request Solution	Past Service Requests (Service Request Array)	String	Service Request History Entity	This function returns the solution name associated with the last created Service Request tied to the customer.
Get Last Service Request Status	Past Service Requests (Service Request Array)	String	Service Request History Entity	This function returns the current status associated with the last created Service Request tied to the customer.
Get Last Service Request Type	Past Service Requests (Service Request Array)	String	Service Request History Entity	This function returns the type associated with the last created Service Request tied to the customer.
Get Number Of Interactions In Past Days	Past Interactions (Interaction Array), Days (Integer)	Integer	Interaction History Entity	This function counts the number of interactions in the past given days.

Table 5–132 (Cont.) RTD_Base_Customer_Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Number Of Opportunities In Past Days	Past Opportunities (Opportunity Array), Days (Integer)	Integer	Opportunity History Entity	This function counts the number of opportunities in the past given days.
Get Number Of Service Requests In Past Days	Past Service Requests (Service Requests Array)	Integer	Service Request History Entity	This function counts the number of service requests in the past given days.
Get Opportunity Product Lines	Past Opportunities (Opportunity Array)	String Array	Opportunity History Entity	This function returns the product lines of all customer opportunities.
Get Product Lines Owned	Purchased Items (Purchased Items Array)	String Array	Purchase History Entity	This function returns product lines of the purchased items associated with the customer.
Get Product List	Active (String)	Product Entity Array	Call Product Choice Group, Group Attributes	This function retrieves an array of products. Used with the Call Product dynamic choice group.
Get Promotion List	None	Promotion Entity Array	Promotions Choice Group, Group Attributes	This function is used to return array of Promotion entity for Promotion dynamic choice. The result is either ranked promotion array supplied as incoming parameter of advisors or array of promotions loaded using data source mapping.
Get Retention Action List	Active (String) Category (String))	Action Entity Array	Retention Actions Choice Group, Group Attributes	This function retrieves an array of retention actions. Used with the Retention actions dynamic choice group.
Get Solution List	Active (String) Product Id (String))	Solution Entity Array	Solutions Choice Group, Group Attributes	This function retrieves an array of Solutions. Used with the solutions dynamic choice group.
Get Specific Choice Likelihood	modelName (String) thisChoiceName (string)	Double	Various Choice Attributes	This function returns likelihood for a given inputted choice that is part of a Choice Model (as opposed to a choice event model). As inputs, the user must pass in the value for the "Name" attribute assigned to the choice and the model name that choice is part of.
Get Top Call Product	None	String	Predictive Summary Choice	This function calls the Select Likely Call Product decision and returns the top ranked call product according to it.
Get Top Interaction Duration	None	String	Predictive Summary Choice	This function calls the Select Likely Interaction Duration decision and returns the top ranked call product according to it.
Get Top Interaction Reason	None	String	Predictive Summary Choice	This function calls the Select Likely Interaction Reason decision and returns the top ranked call reason according to it.
Get Total Amount Spent	Purchased Items (Purchase Items Array)	Double	Purchase History Entity	This function sums up the amount the customer spent so far based on purchase amount of products owned.

Table 5–132 (Cont.) RTD_Base_Customer_Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Total Amount Spent in Last 90 Days	Purchased Items (Purchase Items Array)	Double	Purchase History Entity	This function sums up the amount the customer spent in past 90 days.
Get Up Sell Product List	Product Id (String)	Product Entity Array	Up Sell Offers Choice Group, Group Attributes	This function is used to return array of Product entity for Up Sell dynamic choice. The result is either ranked products array supplied as incoming parameter of advisors or array of up sell products loaded using data source mapping.
Multiply	A (Double) B (Double)	Double	Various objects	This function multiplies the given numbers.
Set Choice Model	Choice Model Name (String) Choice Name (String)	None	Close Session integration point	This function is used to store all of the logic needed to set the Choice Models tied to the Choice Groups.
Set Session Rank Offers	Offers (String array)	None	Various Advisor Integration Points	This procedure is used to parse and set the session Rank Offers attribute before decision selection.

5.10 Batch Use of the Base Customer Service Module

The batch examples provided in this base solution demonstrate a technical implementation of the Oracle RTD platform's batch framework. Logic has been included in this Inline Service to allow for the batch operation of this solution. The following scenarios for batch use are:

- **Batch Scenario 1** - Learning of offline customer call data to update the analytical choice groups, for example, Call Reason, Call Duration, and so on.

  To facilitate this example, source customer data for this example should be provided in the form of a customer text file which contains data that allows for the analytical choice group models to be updated.

- **Batch Scenario 2** - Batch recommendations of actions that customers can use for outbound applications.

  The batch Inline Service should provide batch recommendations and write to an output table where a customer would use the output records for external purposes.

Each of the above use cases is configured through .java files and adheres to the sequence of API's that the batch framework expects, namely:

1. **`init()`** - Called once by the framework before starting the batch's processing loop.
2. **`getNextInput()`** - Returns the next input row to be processed by the batch.
3. **`executeRow()`** - The batch job implements this method to process the input row that was returned by `getNextInput()`.
4. **`flushOutputs()`** - Called by the framework to allow the batch job to flush its output table buffers.
5. **`cleanup()`** - Called by the framework after the batch is finished or is being stopped. Cleans up any resources allocated by the batch job, such as the result set created by its `init()` method.

For full details of the methods of the BatchJob interface and execution of the batch framework, refer to the Oracle RTD Batch Framework chapter in *Oracle Real-Time Decisions Platform Developer's Guide*.

A separate .java file has been created for each of the batch use cases described in this section. These java files are located under the path, `RTD_ILS_HOME/RTD_Base_Customer_Service/src/batch` where *`RTD_ILS_HOME`* is the directory into which the Inline Service RTD_Base_Customer_Service has been saved. The following are the .java files included:

- BatchLearn.java
- BatchRecommend.java

> **Important:** These java files should not be relocated as their path is referenced by the Inline Service when registering each of the batch process to the batch framework.

This section contains the following topics:

- Section 5.10.1, "Registering Batch Jobs for Base Customer Service"
- Section 5.10.2, "BatchLearn.java"
- Section 5.10.3, "BatchRecommend.java"

5.10.1 Registering Batch Jobs for Base Customer Service

The logic for registering each of the batch job .java files can be found in the Logic section of the Application object of the Inline Service. Both .java files are registered against the default path, `RTD_Base_Customer_Service/src/batch`.

If users decide not to include batch functionality with their Inline Service, they must comment out the logic in this section.

5.10.2 BatchLearn.java

The `BatchLearn.java` file contains the logic to process a list of customers and offer responses, obtained through a SQL query, and calls the Inline Service informant to update its learning models.

init()

The `init()` API loads all of the batch parameters defined for this batch job, such as the SQL statement that pulls customer interaction data to be processed by the job. Currently, the java file has a default SQL statement that pulls data from a table called `BatchInteractions`. However, customers should replace this with their own table from which to learn.

getNextInput()

The `getNextInput()` API processes the given result set of customer data and sets the data into the inputRow object to be processed by the `executeRow()` API.

executeRow()

The `executeRow()` API calls the Start Session informant to begin the Oracle RTD session, passing it the appropriate request parameters. It then calls the `CloseSessionInformant()` API, which in turn calls the Close Session integration point.

The Close Session integration point contains the logic that updates all of the learning models tied to the interaction.

flushOutputs()

The `flushOutputs()` API is not used for this batch job.

cleanup()

The `cleanup()` API closes out the result set built from the `init()` API and closes the database connection.

5.10.3 BatchRecommend.java

The `BatchRecommend.java` file contains the logic to process a list of customers obtained through a SQL query, and calls the Inline Service advisor Get Actions to retrieve the top ranked action for a given customer. This output is then written to an output table defined by the customer.

init()

The `init()` API loads all of the batch parameters defined for this batch job, such as the SQL statement for pulling the list of customers for whom to retrieve the best action. In this example, the SQL is tied to a table `BatchCustomers`. However, customers should replace this with their own table from which to predict.

getNextInput()

The `getNextInput()` API processes the given result set of customer data and sets the data into the inputRow object to be processed by the `executeRow()` API.

executeRow()

The `executeRow()` API calls the Get Actions integration point, passing it the appropriate customer id from the inputRow object. The recommended action is then obtained from the response and passed to the `insertOrUpdate()` API, which uses JDBC functionality and prepares the INSERT or UPDATE statement to be used after the `flushOutputs()` API is called.

flushOutputs()

The `flushOutputs()` API is called to execute all of the INSERT and UPDATE statements that have been collected and prepared by the `insertOrUpdate()` API.

cleanup()

The `cleanup()` API closes out the result set built from the `init()` API and closes the database connection.

Configuring the Base Inline Services

This chapter outlines the steps that you can take to configure the Base Inline Services.

As outlined in Part 2, each Base Inline Service contains a template of metadata. This template enables customers to tie their particular front end to a variety of Oracle RTD Integration Points. Oracle RTD can use these Integration Points both for Oracle RTD decisioning and for analysis of the customer web site.

While the Inline Service contains a variety of decisioning and data transformation logic, users must still take this Inline Service and identify what elements are necessary to support their business workflow and requirements.

To achieve this alignment, Oracle recommends the following high-level steps:

1. Align business workflow with Oracle RTD Integration Points and metadata.
   a. Identify and sequence the required Integration Points.
   b. Identify incoming interaction data.
2. Integrate Oracle RTD with the customer front end.
   a. Refer to Part II - Integration with Oracle RTD of the *Oracle Real-Time Decisions Decision Studio Reference Guide* for integration methods to external applications.
3. Map entity attributes to physical data sources.
   a. Add additional entity attributes as required by the business process.
4. Add additional logic as required.
   a. Areas include eligibility rules, additional physical and derived attributes, performance goals and scoring rules, additional Integration Points.
   b. Refer to the following for guidelines:

Oracle RTD platform documentation, which includes the Getting Started tutorial in Part 1 of the *Oracle Real-Time Decisions Decision Studio Reference Guide*

Your third party implementer documentation, if appropriate

6.1 Aligning Your Business Flow with Oracle RTD

Each Base Inline Service includes a collection of integration points that collects current interaction data about the front-end process (informants) and provides real time recommendations and scores (advisors).

In order to incorporate your own customer data into the Oracle RTD models, it is important to select and order how each applicable integration point can be used in your business process. For example, not all of the provided Advisors (such as Get Advertisements, Get Up Sell Offers, and so on) may be applicable in the intended workflow. Furthermore, it is critical to identify which data can be passed through the Oracle RTD integration points in real time in the context of the business workflow.

As an early design task, the Inline Service should be evaluated for what pieces are applicable and what data will be available for building the models, either as current interaction data or as data from data sources. See Section 6.3, "Mapping Entity Attributes to Customer Data Sources" for further discussions on data mapping.

6.2 Integrating Oracle RTD with the Customer Front End

Several methods for integrating your front end with Oracle RTD are available. These include the use of one of the following:.

- Java smart client
- .NET smart client
- Direct web services

Each of these methods is discussed in detail in Part II - Integration with Oracle RTD of the *Oracle Real-Time Decisions Decision Studio Reference Guide*, where you will find steps on completing the integration.

6.3 Mapping Entity Attributes to Customer Data Sources

Each Base Inline Service contains a logical entity model that joins common customer and web attributes that a system may gather in the process of navigating through the business workflow.

As a design exercise, customers who use a Base Inline Service should review the entity structure and map the appropriate attributes to their own data schema. If there are useful data attributes or business objects that are not included in the Base Inline Service entities, customers should feel free to incorporate them into the entity model.

Data that captures the context of the customer interaction can be extremely useful in understanding customer behavior and offer acceptance. In addition to data captured from a back end data schema, users should also evaluate what information can be captured as current interaction data for the web interaction and be sent to Oracle RTD via the integration points.

Customers can refer to the Getting Started tutorial in Part 1 of the *Oracle Real-Time Decisions Decision Studio Reference Guide* for details on adding data sources and modifying or adding entity attributes.

6.4 Adding Additional Logic as Necessary

While each Base Inline Service covers a wide variety of data inputs and touchpoints, it is important to recognize that it serves as a base point. Logic unique to a customer's own business processes may need to be added to basic Oracle RTD operations, such as the following:

- Performing analytics using the Oracle RTD models
- Making recommendations through the Oracle RTD decision engine

Customers may also add custom logic to a Base Inline Service. Examples of customizations can include the following:

- Additional derived data attributes
- Modified scoring methods for performance goals
- Modification of choice event outcomes
- Additional analytical models to learn on unique business processes
- Additional eligibility or filtering rules for applicable choices

For more information on performing customizations to an Inline Service, refer to the following manual:

- *Oracle Real-Time Decisions Decision Studio Reference Guide*

Reference Library

Part 3 describes the component elements of the Reference Library Inline Services and associated elements.

Part 3 contains the following chapters:

Numeric Predictions

This chapter describes the elements in the Numeric_Prediction_ECommerce Inline Service. It contains the following topics:

- Section 7.1, "Introduction to Numeric Predictions"
- Section 7.2, "Application Object"
- Section 7.3, "Performance Goals"
- Section 7.4, "Choice Groups and Choices"
- Section 7.5, "Entities"
- Section 7.6, "Integration Point Workflows"
- Section 7.7, "Informants"
- Section 7.8, "Advisors and Associated Decisions"
- Section 7.9, "Models"
- Section 7.10, "Functions"
- Section 7.11, "Running Numeric Predictions with Loadgen"

7.1 Introduction to Numeric Predictions

When users configure models that are numerically based, users typically specify numeric ranges for each choice so that when the models are set, a discrete numeric value can fall into one of the numeric range choices. Examples of these choices include revenue, call duration, and interaction time, where example choice values are "0-10 Minutes", "10-20 Minutes", "0-100 Dollars", and so on. The creation of choices that group numbers into predefined numeric ranges allows users to develop models that are meaningful to the corresponding workflow for reporting.

Oracle RTD predictions against these ranges can be limiting, however, unless the ranges are small enough to make the predictions also meaningful. The use of numerical predictive techniques in Oracle RTD resolves this issue by allowing users the best of both worlds - that is, to configure both discrete ranges that are significant for reporting purposes as well as to configure choices that can be used to predict an explicit number as opposed to ranges.

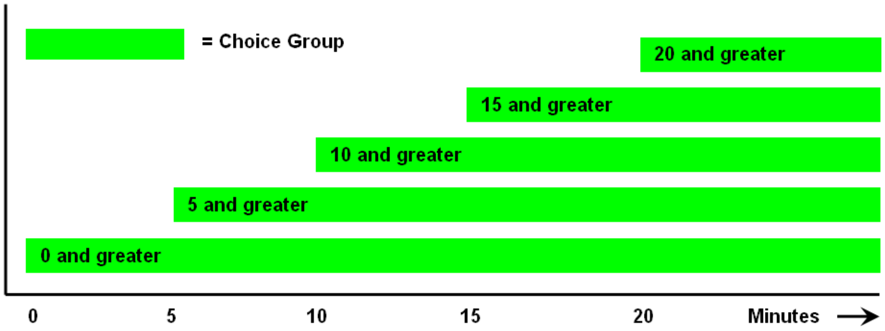
The Numeric_Prediction_ECommerce Inline Service example demonstrates how to apply numerical predictive techniques in the Oracle RTD platform. While the method used is completely general and can be applied to any case where numeric prediction is required, the Numeric_Prediction_ECommerce Inline Service demonstrates specifically how web duration (minutes) can be modeled and predicted.

The example scenario is an application integrated with Oracle RTD which has a requirement to predict web duration. After each user session on the web site, an Oracle RTD model is built that analyzes the total time spent on a web site. As each new visitor connects to the site, at the beginning of the user session, Oracle RTD uses the model and the current user's profile and predicts how long that user will stay on the web site. This can help to determine which message to present to the user. For example, short stay users could be offered a special promotion, while longer stay users could be invited to take a survey.

For predicting numeric values, Oracle RTD utilizes a method based on quantile regression and overlapping threshold techniques. The key component consists of choices whose values are equally spaced, overlapping numeric ranges. Users can configure the ranges as dynamic choices by supplying application parameters. For more details of these parameters, see Section 7.2, "Application Object."

As an example, with the application parameter Min Output set to 0, Spacing set to 5, and Max Output set to 20, the following dynamic choices are configured:

- 0 and greater
- 5 and greater
- 10 and greater
- 15 and greater
- 20 and greater



For each web session, as the session closes, the value for the total duration is passed to Oracle RTD. The Inline Service is configured to update the model for each of the choices that the session duration falls into. So if the session duration is 17 minutes, the choices, "0 and greater", "5 and greater", "10 and greater", and "15 and greater" will all have their associated models updated. This learning pattern is repeated for each session.

When a duration prediction is requested, based on session attributes, Oracle RTD provides a likelihood of occurrence for each of the choices. Oracle RTD then determines how many choices have a likelihood greater than a given minimum value - this minimum value is supplied by users as an application parameter Choice Likelihood Threshold, whose default value is 0.5.

For each choice where the likelihood is above the minimum value, Oracle RTD then applies the formula:

- Prediction value of a numeric target = (Number of the dynamic choices that meet the minimum choice likelihood) * (Spacing between the choices) + (Minimum value for the target number)

Following the example, if four of the choices have a likelihood greater than the Choice Likelihood Threshold, the formula calculates the predicted value as:

- 4 (number of above threshold choices) * 5 (choice spacing) + 0 (minimum value) = 20

From this, it should be clear that the more narrow the spacing, the more precise the numeric prediction can be, assuming that there is enough data to learn on.

Use of Dynamic Choices for Numeric Predictions

Unlike the conventional configuration of dynamic choices, the choice values are not stored in a separate data source and then retrieved when a decision is called for that choice group. Instead, the values for the dynamic choices in the Numeric_Prediction_ECommerce Inline Service are created at run time using predefined java logic with parameters based on user defined application settings. These application parameters are described in Section 7.2, "Application Object," and the functions that create and use these choices are described in Section 7.10, "Functions."

There are two sets of dynamic choices created in the Numeric_Prediction_ECommerce Inline Service, both of which have their parameters defined at the Inline Service Application Object level:

- The first set is used to dynamically create discrete ranges of numerical values, useful for analytics and reporting. The choice group is **Web Site Duration**.
- The second set creates dynamic choices that consist of the overlapping numeric thresholds required for numeric predictions using the quantile regression techniques. As these thresholds overlap with each other, interpreting reports developed for these dynamic choices can be difficult to understand, hence the reason for the discrete ranges as well.

  From a reporting standpoint, users should refer only to the discrete range choices under the Analysis parent choice group for reports. In fact, the choice group created for the overlapping thresholds, **Web Site Duration Prediction**, is hidden from the Decision Center by default to avoid this confusion to the user.

In the Numeric_Prediction_ECommerce Inline Service, there are two Decisioning choice groups, one dynamic and one static:

- The choice group that contains the overlapping dynamic choices is the **Web Site Duration Prediction** choice group, and its associated model provides the likelihood information for the numeric prediction formula.
- The actual numeric prediction is provided by the (static) choice group **Web Duration Choice**.

Using the Base E-Commerce solution as a foundation, the Numeric_Prediction_ECommerce Inline Service shows how the choice group **Web Duration Choice** has been reconfigured with the necessary java logic to implement explicit numeric predictions.

Further details can be found in Section 7.4, "Choice Groups and Choices." Also, Section 7.6, "Integration Point Workflows" provides information on how the choice groups are used in the Numeric_Prediction_ECommerce learning workflow.

7.2 Application Object

At the Inline Service Application object level are the configurations parameters that a user must set to define the limits of the numeric range for the target choice groups, as well as the spacing of the range for both the analytics and predictive portions of the Inline Service. These parameters are referenced in the learning and decisioning code found in the Inline Service. Refer to Section 7.10, "Functions" for further details.

Note: Do not alter the names of the Application Parameters, as they are directly referenced by several functions in the Numeric_Prediction_ECommerce Inline Service example.

Table 7–1 Application Parameters for Numeric Predictions E-Commerce Inline Service

Name	Type	Default Value	Description
Analysis Bucket Boundary Array	Integer Array	(0, 1, 5, 10, 20, 30)	When defined, the standard spacing indicated by the application parameter Spacing For Analysis will not be used to create the numeric range for each dynamic choice. Instead, the user can define the spacing used for analytics through this parameter.
Choice Likelihood Threshold	Double	0.5	Defines the minimum likelihood value to determine if a choice is above the confidence threshold and is to be applied to the numeric predictions formula. See the Get Numeric Prediction function.
Max Output	Integer	30	Defines the maximum numeric value for the target choice group used for prediction.
Max Output for Analysis	Integer	Application/ Max Output	Defines the maximum numeric value for the target choice group used for analysis.
Min Output	Integer	0	Defines the minimum numeric value for the target choice group used for prediction.
Min Output for Analysis	Integer	Application/ Min Output	Defines the minimum numeric value for the target choice group used for analysis.
Spacing	Integer	2	Defines the spacing for each dynamic choice created under the prediction choice group. Note: The smaller the spacing the larger the number of dynamic choices there will be. This will require more occurrences for the model to learn for each group but allows for a more refined prediction. Larger spacing may result in faster learning but less refined predictive outcomes.
Spacing for Analysis	Integer	Application/ Spacing	Defines the spacing for each dynamic choice created under the analysis prediction choice group.
Target Choice Base Name	String	"minutes"	Describes the numerics being modeled and will also be included in the name of the dynamic choice.
Target Choice Group Name for Analysis	String	"WebSiteDurationCG"	Object name of the choice group where the dynamic choices used for analysis will be created.
Target Choice Group Name for Prediction	String	"WebSiteDurationPrediction"	Object name of the choice group where the dynamic choices used for prediction will be created.

7.3 Performance Goals

One performance goal, Maximize Likelihood, is included in the Numeric_Prediction_ECommerce Inline Service, to be used by the loadgen script.

7.4 Choice Groups and Choices

Analysis and Decisioning Based Choice Groups

Although the Numeric_Prediction_ECommerce Inline Service demonstrates numeric predictions for one subject area only, namely Web Duration, two types of choice group have been configured, one for analysis and one for prediction. Each is tied to a model which analyses web duration, but they differ in the way that the dynamic choices are created for each group.

For more useful analysis reports, numeric ranges represented in the dynamic choices should be bounded. For example, for a choice group where users want to model a numeric range of 0 to 100, with a spacing of 10 (see the parameters for the Application Object), the Inline Service creates dynamic choices for analysis as follows:

- 0-10
- 10-20
- 20-30

and so on up to 90-100. This allows for discrete reports to be seen in Decision Center, where users can go to one specific choice to look at the reports depending on the desired range.

For predictions, the quantile regression method relies on overlapping buckets of numeric ranges to be modeled, where a median number is determined based on the most likely buckets for any given prediction. For the same example range of 0-100, an overlapping scheme with a spacing of 10 results in dynamic choice groups with the following values:

- "0 and above"
- "10 and above"
- "20 and above"

up to "90 and above." So, if the session resulted in a number of 25 to be modeled, the choice groups, "0 and above," "10 and above," and "20 and above" will each be updated accordingly.

With these dynamic choices of overlapping buckets, a prediction algorithm can then be applied based on the most likely choices to determine the final numeric value. In the Numeric_Prediction_ECommerce Inline Service, this is achieved by the Get Numeric Prediction function.

While the resulting Decision Center reports still provide useful information, users may find it more useful to view the reports in bounded ranges as it provides simply one "location" to find the report they want.

The following choice groups are configured in the Inline Service Numeric_Prediction_ECommerce:

- Analysis - Web Site Duration
- Decisioning - Web Duration Choice
- Decisioning - Web Site Duration Prediction

Note: For more information about how the choice groups are used in the Inline Service Numeric_Prediction_ECommerce, see Section 7.6, "Integration Point Workflows."

7.4.1 Analysis - Web Site Duration

Table 7–2 describes the configuration parameters for the Analysis - Web Site Duration choice group.

Table 7–3 provides details of the dynamic choice parameters for this choice group.

Table 7–2 Configuration Parameters for Analysis - Web Site Duration Choice Group

Parameter	Description
Choice Group Name	Analysis - Web Site Duration
Choice Attributes	Numeric Prediction Choice Attr - The Type of this attribute is Numeric Prediction Entity. - This attribute has no Value - it is a placeholder required for configuration. Likelihood - The Type of this attribute is Double. - The Value is predicted by the Web Site Duration model.
Scores	For the performance goal Maximize Likelihood, score is determined by choice Likelihood attribute from the corresponding choice event model.
Choice Events	None
Choice Eligibility	None
Group Attributes	Numeric Prediction Group Attr. - The Type is Numeric Prediction Entity. - There is no Value as this entity is populated via the Learning by bucketing function.
Group Eligibility	None
Dynamic Choices	For details, see Table 7–3.

Table 7–3 Dynamic Choice Details for Analysis - Web Site Duration Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Numeric Prediction Group Attr
Choice attribute to assign the entity data	Numeric Prediction Choice Attr
Entity attribute that contains the choices id	Choice id
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

7.4.2 Decisioning - Web Duration Choice

This choice group is configured to return the numeric predicted value for the Web Duration choice. Instead of a choice id being returned, the value of the Web duration attribute will be an entity of the choice Web Duration configured under this choice group.

Table 7–4 describes the configuration parameters for the Decisioning - Web Duration Choice choice group.

Table 7–4 Configuration Parameters for Decisioning - Web Duration Choice Choice Group

Parameter	Description
Choice Group Name	Decisioning - Web Duration Choice
Choice Attributes	Web Duration ■ The Type of this attribute is Double. ■ The Value is determined by the function Get Numeric Prediction, with the inputs Application / Target Choice Group Name for Prediction and Model Name = Web Site Duration Prediction Model.
Scores	For the performance goal Maximize Likelihood, score is 0.
Choice Events	None
Choice Eligibility	None
Group Attributes	None.
Group Eligibility	None
Dynamic Choices	None

7.4.3 Decisioning - Web Site Duration Prediction

Table 7–5 describes the configuration parameters for the Decisioning - Web Site Duration Prediction choice group.

Table 7–6 provides details of the dynamic choice parameters for this choice group.

Table 7–5 Configuration Parameters for Decisioning - Web Site Duration Prediction Choice Group

Parameter	Description
Choice Group Name	Decisioning- Web Site Duration Prediction
Choice Attributes	Numeric Prediction Choice Attr ■ The Type of this attribute is Numeric Prediction Entity. ■ This attribute has no Value - it is a placeholder required for configuration. Likelihood ■ The Type of this attribute is Double. ■ The Value is predicted by the Web Site Duration Prediction model.
Scores	For the performance goal Maximize Likelihood, score is determined by choice Likelihood attribute from the corresponding choice event model.
Choice Events	None
Choice Eligibility	None

Table 7–5 (Cont.) Configuration Parameters for Decisioning - Web Site Duration Prediction Choice Group

Parameter	Description
Group Attributes	Numeric Prediction Group Attr ■ The Type is Numeric Prediction Entity. ■ It is populated through the function Get Group Attr Array, with the input Target Choice Base Name from the value of Application / Target Choice Base Name.
Group Eligibility	None
Dynamic Choices	For details, see Table 7–6.

Table 7–6 Dynamic Choice Details for Decisioning - Web Site Duration Prediction Choice Group

Parameter	Value
Group attribute containing the list of entities for choices	Numeric Prediction Group Attr
Choice attribute to assign the entity data	Numeric Prediction Choice Attr
Entity attribute that contains the choices id	Choice id
Distribution mode for choices over choice group folders	Spill
Maximum number of choices in one Decision Center folder	100

7.5 Entities

This section outlines the entity model used for the Numeric_Prediction_ECommerce Inline Service. It is a subset of the entity model used on the RTD_Base_ECommerce Inline Service with only the portions required to articulate the Web Site Duration use case.

Several session driven attributes are included in this example for simulation purposes, to be used in conjunction with the loadgen script included. These attributes do not reflect which attributes are to be used for numeric predictions, but are included so that users can see outputs in the Decision Center when running this example.

The following entities are configured in the Numeric_Prediction_ECommerce Inline Service:

- Customer

  The Customer entity is used to store customer data created during numeric prediction simulations through load generator.

- Current Web Interaction

  This entity is used to store web interaction data created during numeric prediction simulations through load generator.

- Numeric Prediction

  This is a placeholder entity used for creating dynamic choices against the web duration choice groups.

7.5.1 Customer

The Customer entity has the following attributes:

- Address State Province

- Age
- Days To Due Date
- Gender
- Industry
- Marital Status
- Minimum Amount Due
- Number Of Children
- Occupation
- Signed Up For EPay
- Tenure

7.5.2 Current Web Interaction

The Current Web Interaction entity has the following attributes:

- Origin To Website
- Time Of Day
- Total Duration In Minutes
- Web User Location

7.5.3 Numeric Prediction

The Numeric Prediction entity has no attributes configured.

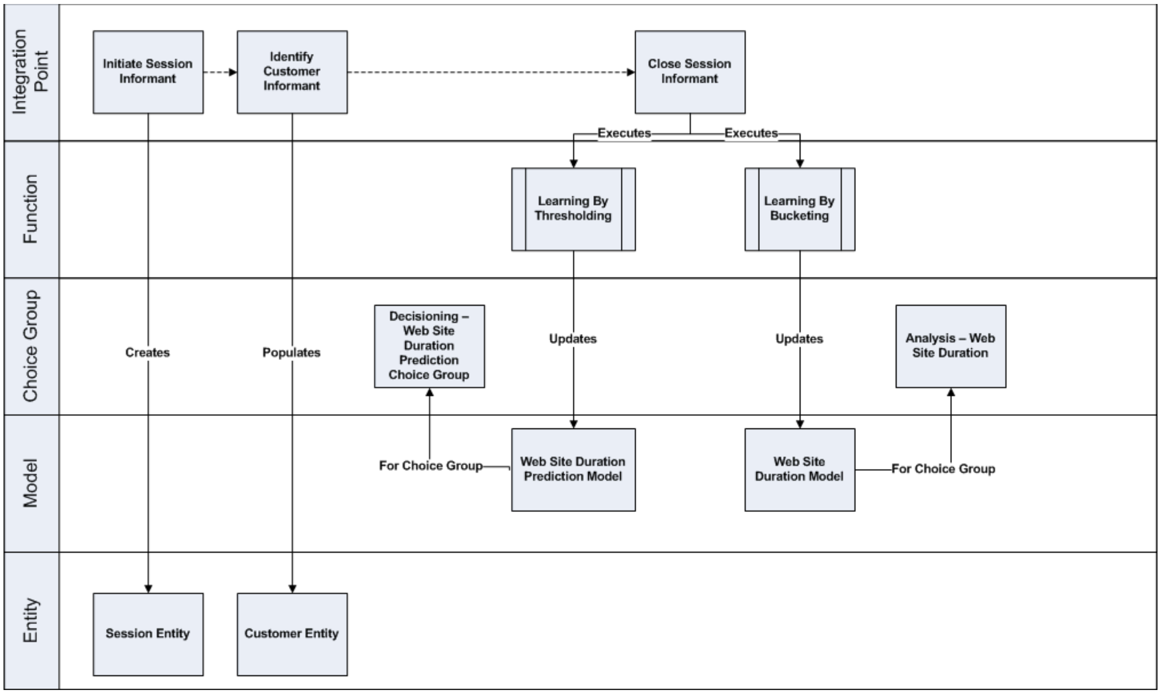
7.6 Integration Point Workflows

This section shows the workflows for the Numeric_Prediction_ECommerce Inline Service integration points and the Inline Service elements used during the processing of the integration points.

Figure 7–1 shows the workflow and elements associated with each informant in the Numeric_Prediction_ECommerce Inline Service. Note that the "learning" logic is included as part of the Close Session informant processing, which updates the following models:

- The model that learns for numeric prediction against the Web Site Duration Prediction choice group
- The model that learns for analysis against the Web Site Duration choice group

Figure 7–1 Numeric_Prediction_ECommerce Learning Workflow



The processing for each informant shown in Figure 7–1 is as follows:

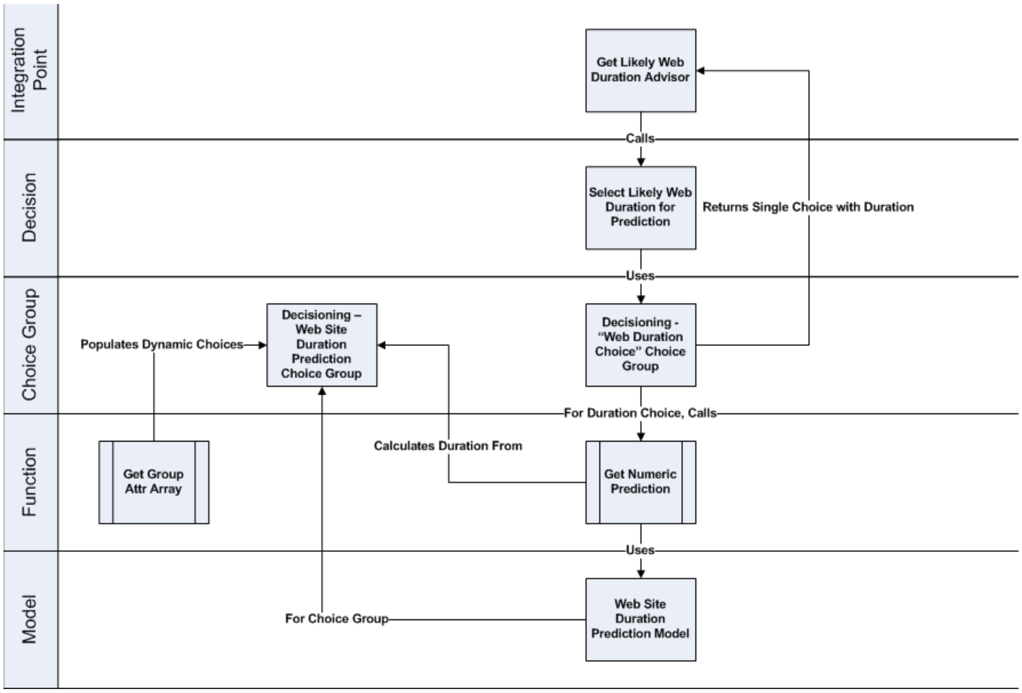
- The Initiate Session informant creates the Session entity
- The Identify Customer informant populates the Customer entity
- The Close Session informant executes the following functions:
  - Learning by Thresholding

    This passes the web duration to the Web Site Duration Prediction Model, which learns for numeric predictions against the Decisioning choice group Web Site Duration Prediction
  - Learning by Bucketing

    This passes the web duration to the Web Site Duration Model, which learns for analysis against the Analysis choice group Web Site Duration

Figure 7–2 shows the workflow for the advisor Get Likely Web Duration in the Numeric_Prediction_ECommerce Inline Service.

Figure 7–2 Numeric_Prediction_ECommerce Advisor Workflow


Integration Point
Decision
Choice Group
Function
Model
Get Likely Web Duration Advisor
Calls
Select Likely Web Duration for Prediction
Returns Single Choice with Duration
Uses
Decisioning - "Web Duration Choice" Choice Group
Populates Dynamic Choices
Decisioning – Web Site Duration Prediction Choice Group
For Duration Choice, Calls
Calculates Duration From
Get Group Attr Array
Get Numeric Prediction
Uses
For Choice Group
Web Site Duration Prediction Model

The processing behind the Get Likely Web Duration advisor shown in Figure 7–2 is as follows:

- The Get Likely Web Duration advisor calls the decision Select Likely Web Duration for Prediction, which uses the Decisioning choice group Web Duration Choice
- The Decisioning choice group Web Duration Choice calls the function Get Numeric Prediction

  The function Get Numeric Prediction calculates the numeric prediction against the Decisioning choice group Web Site Duration Prediction for the model Web Site Duration Prediction Model

  The Decisioning choice group Web Site Duration Prediction has its dynamic choices populated by the function Get Group Attr Array
- The Decisioning choice group Web Duration Choice receives the numeric prediction for the web duration and passes it to the Get Likely Web Duration advisor

7.7 Informants

The following informants are configured in the Numeric_Prediction_ECommerce Inline Service:

- Initiate Session
- Identify Customer
- Close Session

In the Numeric_Prediction_ECommerce Inline Service, these informants together with the advisor Get Likely Web Duration form a subset of the integration points found in the Base E-Commerce solution. They are used by the included loadgen scripts to create the simulated learnings that demonstrate the numeric prediction capabilities.

Note: For more details about the Integration Point processing in the Numeric_Prediction_ECommerce Inline Service, see Section 7.6, "Integration Point Workflows."

7.7.1 Initiate Session

The Initiate Session informant is called to initiate the Oracle RTD session and sends initial information regarding the customer interaction.

Table 7–7 describes the parameters for the Initiate Session informant.

Table 7–7 Informant Initiate Session

Parameter	Description
Informant Name	Initiate Session
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Time of Day (String) mapped to Current Web Interaction.Time Of Day User Location (String) mapped to Current Web Interaction.Web User Location Web Origin (String) mapped to Current Web Interaction.Origin To Website
External System	Web ECommerce
Order	1
Force session close	No
Logic	None
Pre-condition	None

7.7.2 Identify Customer

The Identify Customer informant is called to send information regarding the customer interaction that is collected during the web interaction.

Table 7–8 describes the parameters for the Identify Customer informant.

Table 7–8 Informant Identify Customer

Parameter	Description
Informant Name	Identify Customer
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Address State Province (String) mapped to Customer.AddressStateProvince Customer Id (String) mapped to Customer.Customer Id Gender (String) mapped to Customer.Gender Industry (String) mapped to Customer.Industry Marital Status (String) mapped to Customer.Marital Status
External System	Web ECommerce
Order	5
Force session close	No

Table 7–8 (Cont.) Informant Identify Customer

Parameter	Description
Logic	None
Pre-condition	None

7.7.3 Close Session

The Close Session informant is called to close out the session and send in the final web duration value. It also contains the logic which calls the functions that update the appropriate dynamic choice models for numeric prediction.

Table 7–9 describes the parameters for the Close Session informant.

Table 7–9 Informant Close Session

Parameter	Description
Informant Name	Close Session
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Web Duration (Integer)
External System	Web ECommerce
Order	100
Force session close	Yes
Logic	Contains logic that calls the functions Learning By Thresholding and Learning By Bucketing. These functions update the choices groups for Web Duration in both the Analysis and Decisioning choice groups.
Pre-condition	None

7.8 Advisors and Associated Decisions

The following advisor is configured in the Numeric_Prediction_ECommerce Inline Service:

- Get Likely Web Duration

In the Numeric_Prediction_ECommerce Inline Service, the advisor Get Likely Web Duration together with the three informants Initiate Session, Identify Customer, and Close Session, form a subset of the integration points found in the Base E-Commerce solution. They are used by the included loadgen scripts to create the simulated learnings that demonstrate the numeric prediction capabilities.

> **Note:** For more details about the Integration Point processing in the Numeric_Prediction_ECommerce Inline Service, see Section 7.6, "Integration Point Workflows."

7.8.1 Get Likely Web Duration

The advisor Get Likely Web Duration returns a single choice, Web Duration, whose attribute Web Duration contains the predicted numeric value for web duration for the given session inputs.

Table 7–10 describes the parameters for the advisor Get Likely Web Duration.

Table 7–10 Advisor Get Likely Web Duration

Parameter	Description
Advisor Name	Get Likely Web Duration
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Address State Province (String) mapped to Customer.AddressStateProvince Customer Id (String) mapped to Customer.Customer Id Gender (String) mapped to Customer.Gender Industry (String) mapped to Customer.Industry Marital Status (String) mapped to Customer.Marital Status
External System	CRM System
Order	0
Force session close	No
Decision	Select Likely Web Duration for Prediction
Group Decision	Select Likely Web Duration for Prediction
Default Choices	None
Logic	None
Pre-condition	None

Table 7–11 describes the parameters for the decision for the advisor Get Likely Web Duration.

Table 7–11 Decision for Advisor Get Likely Web Duration

Parameter	Description
Decision Name	Select Likely Web Duration for Prediction
Select Choices From	Web Duration Choice
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

7.9 Models

Two models are included in the Numeric_Prediction_ECommerce Inline Service. The first model, Web Site Duration Model, is used for the analysis-based choice group while the second model, Web Site Duration Prediction Model, is used for the prediction based model.

Note: For more information about how the models are used in the Numeric_Prediction_ECommerce Inline Service, see Section 7.6, "Integration Point Workflows."

7.10 Functions

The functions included in Numeric_Prediction_ECommerce Inline Service are used for learning the dynamic choice buckets and for applying the scoring algorithm for predicting a number.

Table 7–12 Numeric_Prediction_ECommerce Functions

Function	Inputs	Outputs	Comments
Get Group Attr Arrays	Target Choice Base Name	Numeric Prediction Entity Array	Forms the dynamic choice ids for a given choice group name. It is called when dynamic choices are used for prediction. This function is used for the prediction (thresholding) dynamic choices in the Web Site Duration Prediction choice group which are used for numeric prediction. The dynamic choice id format must match the format used in the function Learning By Thresholding, which record choices in that choice group in the model Web Site Duration Prediction Model.
Get Numeric Prediction	Choice Group Id Model Name	Double	This function is used to calculate numeric prediction against the Web Site Duration Prediction choice group for the model Web Site Duration Prediction Model. NOTE: The algorithm used to calculated the number is specifically tied to the use of the overlapping choice groups for the Web Site Duration Prediction choice group and should not be altered.
Get Prediction	Choice Group Id	Double	This function is used to calculate numeric prediction against the Web Site Duration Prediction choice group.
Learning By Bucketing	Target Number Model Name	None	This function generates the dynamic choices based on non-overlapping buckets for analysis corresponding to the given Target Number and records the dynamic choices to a choice model with the given Model Name. [The Target Number is the value of the numeric to be modeled. In this Inline Service, the Target Number is the value for the Web Duration as passed to the Close Session informant.] The value of the dynamic choice to update is determined by the application parameters and the incoming target number. This function is called by the Close Session informant to enable learning by bucketing, which aims at presenting easy-to-interpret results on decision center. If the application parameter Analysis Buckets Boundary Array has valid values, the buckets can have unequal widths, determined by an array of boundaries given by users. Otherwise, by default, the buckets have equal widths, determined by the min, max and spacing, given by users according to their prior knowledge set through Application parameters.
Learning By Thresholding	Target Number Model Name	None	This function records dynamic choices corresponding to the given Target Number to a choice model with the given Model Name. [The Target Number is the value of the numeric to be modeled. In this Inline Service, the Target Number is the value for the Web Duration as passed to the Close Session informant.] The dynamic choices created and updated from this function involve overlapping buckets, and are determined by the application parameters. These overlapping bucket choices are then used to determine the numeric prediction by the Get Numeric Prediction function. This function is called by the Close Session informant to enable learning by thresholding, which aims at calculating numeric prediction. The min, max and spacing are given by users according to their prior knowledge through Application parameters.

Note: For more information about how the functions are used in the Numeric_Prediction_ECommerce Inline Service, see Section 7.6, "Integration Point Workflows."

7.11 Running Numeric Predictions with Loadgen

To demonstrate the learning and prediction functions of this Inline Service, an XML script has been included with the Inline Service Numeric_Prediction_ECommerce, to be run using the Oracle RTD loadgen tool. This script, `NumericP.xml`, can be found in the `etc` folder of the main Inline Service project folder, for example, `ORACLE_RTD_STUDIO_HOME\Numeric_Prediction_ECommerce\etc\NumericP.xml`.

In addition to this file, the `etc\data` folder contains the necessary data simulation files that are used by the script.

After the XML script is opened in the loadgen tool, the path to the data files must be changed to point to the directory structure of where the data file is located:

1. After loading the XML script in loadgen, ensure that the client configuration file parameter is pointing to the right properties file, for example, `clientHttpEndPoints.properties`.
2. Navigate to the Edit script tab.

   For each of the three integration points, modify the Input File URL to reflect the path to where the data text file is located, for example, `C:/RTD_Studio/Numeric_Prediction_ECommerce/etc/data/NumericP.txt`.
3. Run the simulation.

After the simulation is completed, you can navigate to the Decision Center and review how the dynamic choices are set up.

Batch Processing

This chapter describes the elements in and associated with the CrossSellBatch Inline Service. It contains the following topics:

- Section 8.1, "Introduction to Batch Processing"
- Section 8.2, "Setup of the Sample Batch Data"
- Section 8.3, "The Batch .java Files"
- Section 8.4, "Batch Processing Inline Service Elements"
- Section 8.5, "Running Batch Processing"

8.1 Introduction to Batch Processing

The CrossSellBatch Inline Service is a technical example of the use of the batch framework that is released with the Oracle RTD platform. It is based on the Cross Sell example Inline Service that is released with Oracle RTD. Refer to *Oracle Real-Time Decisions Installation and Administration Guide* and *Oracle Real-Time Decisions Platform Developer's Guide* for additional information on both the original Cross Sell Inline Service and the batch frame work. The CrossSellBatch Inline Serviceshould be used as a reference for developers when adding batch functionality to their own Inline Services.

In this example, batch processes have been created to support a workflow process in which Oracle RTD is used first to recommend offers to customers, and then to learn on the customer response, as follows:

- One batch process reads through a list of customers and writes to an output table the best recommended offer for each customer.
- The next batch process simulates the customer response to the recommended offer and then writes the result to a response table.

- After this is done, a final batch process reads the response table and calls Oracle RTD to update the models with the response data.

In addition to the batch process that recommends the best offer, the CrossSellBatch Inline Service also contains an example of scoring customers against a given offer that is specified by the user.

8.2 Setup of the Sample Batch Data

A sample data schema and test data to be used with this example is included with the CrossSellBatch Inline Service. Setup of this data is required in order to test any of the batch processes included in this Inline Service as the batch processes rely on the created data sources.

All files for creating the database tables and loading data can be found directly in the Inline Service project folder under the `/etc` folder. To create the schema and populate the batch tables., you must execute the `InitAppDB.cmd` that is appropriate to your database type through a command line.

For example, for SQL Server, the files can be found under the path, *`RTD_ILS_HOME`*`/CrossSellBatch/etc/data/SQLServer`, where *`RTD_ILS_HOME`* is the directory into which the CrossSellBatch Inline Service was saved during installation.

Each `InitAppDB` command has the following structure:

```
InitAppDB sdroot host port database runtimeUser adminUser [adminPassword]
```

Table 8–1 describes the parameters for the `InitAppDB` script.

Table 8–1 Parameters for InitAppDB Script

Parameter	Description
`sdroot`	The full path of the directory where the Oracle Real-Time Decisions files are installed.
`host`	The name of the computer hosting the database server. If you installed your Oracle RTD Database on a SQL Server named instance, specify *`host\instance_name`*.
`port`	The database port number.
`database`	The name of the database, or for Oracle Database, the SID.
`runtimeUser`[1]	The user name of the run-time user for the system.
`adminUser`	The name of a user that has rights to create tables and stored procedures on the database.
`adminPassword`	The password of the administrative user.

[1] For Oracle Database, the *`runtimeUser`* and *`adminUser`* are the same user.

8.3 The Batch .java Files

The CrossSellBatch Inline Service demonstrates a technical implementation of Oracle RTD Platform's batch framework, and provides coding for the following use case scenarios:

1. **Batch Learning** - Given a set of customer and offer data, through the batch process, call an Oracle RTD informant to process event learning of offer responses.
2. **Batch Learning with Direct Event Recording** - Given a set customer and offer data, update event models directly via java code.

3. **Batch Offer Selection** - Given a set of customer data, perform a batch advisor call to record the best offer per customer and record results to an output table.
4. **Batch Customer Scoring** - Given a specific offer, process a customer list and provide a score for that offer for each customer.
5. **Offer Acceptance ETL** - Through the batch process, demonstrate an ETL (Extract, Transform, and Load) process where simulated customer responses are recorded to an Oracle RTD table for future learning using standard JDBC java functionality.
6. **Offer Acceptance ETL with the RTD Table Inserter** - Through the batch process, demonstrate an ETL process where simulated customer responses are recorded to an Oracle RTD table for future learning using Oracle RTD Table Inserter APIs.

Each of these use cases is configured through java classes that implement the batch framework BatchJob interface, namely:

1. **`init()`** - Called once by the framework before starting the batch's processing loop.
2. **`getNextInput()`** - Returns the next input row to be processed by the batch.
3. **`executeRow()`** - The batch job implements this method to process the input row that was returned by `getNextInput()`.
4. **`flushOutputs()`** - Called by the framework to allow the batch job to flush its output table buffers.
5. **`cleanup()`** - Called by the framework after the batch is finished or is being stopped. Cleans up any resources allocated by the batch job, such as the result set created by its `init()` method.

For full details of the methods of the BatchJob interface and execution of the batch framework, refer to the Oracle RTD Batch Framework chapter in *Oracle Real-Time Decisions Platform Developer's Guide*.

A separate .java file has been created for each of the batch use cases described in this section. These java files are located under the path, `<RTD Studio Workspace>/CrossSellBatch/src/crosssell/batch` where <RTD Studio Workspace> is the directory into which the CrossSellBatch Inline Service has been saved. The following are the .java files included:

1. LearningBatchJob.java
2. LearningWithDirectEventRecordingBatchJob.java
3. OfferAcceptanceETLBatchJob.java
4. OfferAcceptanceETLTableInserterBatchJob.java
5. OfferScoringBatchJob.java
6. OfferScoringBatchJobReadahead.java
7. OfferSelectBatchJob.java

Important: These java files should not be relocated as their path is referenced by the Inline Service when registering each of the batch process to the batch framework.

Refer to Section 8.4.1, "Application Object" for further information about registering each batch process.

8.3.1 LearningBatchJob.java

The `LearningBatchJob.java` file contains the logic to process a list of customers and offer responses, obtained through a SQL query, and calls the Inline Service informant to update its learning models.

init()

The `init()` API loads all of the batch parameters defined for this batch job, such as the SQL statement for pulling customer information from the `CrossSellBatchLearnings` table.

getNextInput()

The `getNextInput()` API processes the given result set of customer data and sets the data into the inputRow object to be processed by the `executeRow()` API.

executeRow()

The `executeRow()` API calls the Call Start informant to begin the Oracle RTD session, passing it the appropriate request parameters. It then calls the `recordEvent()` API, which in turn calls the Record Events Integration point.

flushOutputs()

The `flushOutputs()` API is not used for this batch job.

cleanup()

The `cleanup()` API closes out the result set built from the `init()` API and closes the database connection.

8.3.2 LearningWithDirectEventRecordingBatchJob.java

For this batch job, given a set customer and offer data, the logic updates event models directly via java code.

The `LearningWithDirectEventRecordingBatchJob.java` file is an extension of `LearningBatchJob.java`. All of the expected APIs for the batch framework are consumed from `LearningBatchJob.java` with the exception of `recordEvents()`, which is overridden here.

recordEvents()

Overrides the `recordEvents()` API from `LearningBatchJob.java`. The `recordEvents()` API updates the choice event model for this Inline Service directly, instead of calling the Record Events Integration point.

8.3.3 OfferAcceptanceETLBatchJob.java

This batch job demonstrates an example ETL process where customer offers are read from the `CrossSellBatchBestOffer` table, and simulated responses are then recorded to the Oracle RTD table, `CrossSellBatchLearnings`, for future learning using standard JDBC java functionality.

init()

The `init()` API loads all of the batch parameters defined for this batch job, and prepares all of the SQL statements to be used for this job.

getNextInput()

The `getNextInput()` API processes the given result set of customer data, and sets the data into the inputRow object to be processed by the `executeRow()` API.

executeRow()

The `executeRow()` API simulates customer responses to the offer contained in the Input Row object. The customer information, offer id, and response are then passed to an `insert()` API to prepare an `INSERT` statement for that data to be used when `flushOutputs()` is called by the framework.

flushOutputs()

The `flushOutputs()` API is called to execute all of the `INSERT` statements that have been collected and prepared by the `insert()` API.

cleanup()

The `cleanup()` API closes out the result set built from the `init()` API, and closes the database connection.

8.3.4 OfferAcceptanceETLTableInserterBatchJob.java

This batch job is functionally the same as **OfferAcceptanceETLBatchJob**, but instead of using JDBC java functionality to write the output to the `CrossSellBatchLearnings` table, the Oracle RTD platform's `TableInserter()` API methods are used instead.

This .java file extends `OfferAcceptanceETLBatchJob.java`.

init()

Overrides the `init()` API from `OfferAcceptanceETLBatchJob.java` and instantiates the `TableInserter()` API.

insert()

Overrides the `insert()` API from `OfferAcceptanceETLBatchJob.java` and sets the INSERT record using the `tableInserter.put()` API.

flushOutputs()

Overrides the `flushOutputs()` API from `OfferAcceptanceETLBatchJob.java` and calls the `tableInserter.flush()` method.

cleanup()

Overrides the `cleanup()` API from `OfferAcceptanceETLBatchJob.java` and closes out the job using the `tableInserter` APIs.

8.3.5 OfferScoringBatchJob.java

For a given offer id, this batch job requests the offer score for a list of customers pulled from the `CrossSellBatchCustomers` data source. The scores are then written to the output table, `CrossSellBatchScores`.

init()

The `init()` API loads all of the batch parameters defined for this batch job and prepares the required SQL statements. It also sets a choice object for the requested offer to be scored, set by the offerId batch job parameter.

getNextInput()

The `getNextInput()` API processes the given result set of customer data, and sets the data into the inputRow object to be processed by the `executeRow()` API.

executeRow()

Using the choice object set in the `init()` API, the `executeRow()` API sets the session() with customer Id from the inputRow object, and then retrieves the likelihood score for the Purchased event for the desired offer.

The likelihood score, along with the customer information is then passed to the `insertOrUpdate()` API, which uses JDBC functionality and prepares the INSERT or UPDATE statement to be used after the `flushOutputs()` API is called.

flushOutputs()

The `flushOutputs()` API is called to execute all of the INSERT or UPDATE statements that have been collected and prepared by the `insertOrUpdate()` API.

cleanup()

The `cleanup()` API closes out the result set built from the init() API and closes the database connection.

8.3.6 OfferScoringBatchJobReadahead.java

The **OfferScoringBatchJobReadahead** batch job has the same scoring functionality as the `OfferScoringBatchJob.java` file, in that it will request a likelihood score for a given offer id for each customer.

The difference is in how the customer data is fed to the RTD server. Instead of passing one customer id at a time to Oracle RTD and then retrieving its offer score, the **OfferScoringBatchJobReadahead** batch job passes a group of customer ids all together to Oracle RTD.

Using this strategy, Oracle RTD caches the session entities ahead of the scoring request. This enables faster processing cycles for users that are dealing with large volumes of customer data.

As with the **OfferScoringBatchJob** batch job, the scores retrieved are then written to the output table, `CrossSellBatchScores`.

init()

The `init()` API loads all of the batch parameters defined for this batch job and prepares the SQL statements required. It also sets a Choice object for the requested offer that is to be scored, set by the offerId batch job parameter.

getNextInput()

The `getNextInput()` API processes the given result set of customer data and sets the data into the inputRow object to be processed by the `executeRow()` API. Rather than process one customer id at a time, it will put together a group of customer id's, enabling Oracle RTD to cache the session entities for each id ahead of calling the `executeRow` method. This is passed to Oracle RTD via the `fillAndCache` method for the Customer entity.

executeRow()

Using the choice object set in the `init()` API, the `executeRow()` API sets the `session()` with the customer Id from the inputRow object and then retrieves the likelihood score for the event Purchased for the desired offer. The likelihood score, along with the customer information is then passed to the `insertOrUpdate()` API , which uses JDBC functionality and prepares the `UPDATE` or `INSERT` statement to be used once the `flushOutputs()` API is called.

flushOutputs()

The `flushOutputs()` API is called to execute all of the INSERT and UPDATE statements that have been collected and prepared by the `insertOrUpdate()` API.

cleanup()

The `cleanup()` API closes out the result set built from the `init()` API and closes the database connection.

8.3.7 OfferSelectBatchJob.java

This batch job takes a given set of customer data and requests the best offers from Oracle RTD using the Offer Request advisor integration point. The results are then written to the output table, `CrossSellBatchBestOffer`.

init()

The `init()` API loads all of the batch parameters defined for this batch job, such as the SQL statement for pulling customer information from the `CrossSellBatchCustomers` table.

getNextInput()

The `getNextInput()` API processes the given result set of customer data and sets the data into the inputRow object to be processed by the `executeRow()` API.

executeRow()

The `executeRow()` API calls the Offer Request integration point, passing it the appropriate customer id from the inputRow object. The recommended offer is then obtained from the response and passed to the `insertOrUpdate()` API, which uses JDBC functionality and prepares the INSERT or UPDATE statement to be used after the `flushOutputs()` API is called.

flushOutputs()

The `flushOutputs()` API is called to execute all of the INSERT and UPDATE statements that have been collected and prepared by the `insertOrUpdate()` API.

cleanup()

The `cleanup()` API closes out the result set built from the `init()` API and closes the database connection.

8.4 Batch Processing Inline Service Elements

This section describes the elements of the CrossSellBatch Inline Service. It contains the following topics:

- Section 8.4.1, "Application Object"
- Section 8.4.2, "Performance Goals"
- Section 8.4.3, "Decisions"
- Section 8.4.4, "Choice Groups and Choices"

- Section 8.4.5, "Entities"
- Section 8.4.6, "Informants"
- Section 8.4.7, "Advisors"
- Section 8.4.8, "Models"
- Section 8.4.9, "Functions"

8.4.1 Application Object

At the Inline Service Application object level, in the Logic section, each of the batch jobs is registered with the batch framework. Registering each of the batch .java files is all that is required to "batch enable" an Inline Service.

The general java syntax for batch job registration is as follows:

```
BatchAgent batchAgent = getBatchAgent();

batchAgent.registerBatch("<Batch Job Identifier Name>",
              "<batch java file src path and name>",
               <batch java file name>.description,
               <batch java file name>.paramDescriptions,
               <batch java file name>.paramDefaults);

batchAgent.registerDataSource("SDDS");
```

For example, to register the Learning batch job and Offer Select batch job with the Batch Framework, the syntax is as follows:

```
BatchAgent batchAgent = getBatchAgent();

/*
 * CrossSellLearning Batch
 */

batchAgent.registerBatch("CrossSellLearning",
                 "crosssell.batch.LearningBatchJob",
                 LearningBatchJob.description,
                 LearningBatchJob.paramDescriptions,
                 LearningBatchJob.paramDefaults);

/*
* CrossSellSelectOffers Batch
 */
batchAgent.registerBatch("CrossSellSelectOffers",
                         "crosssell.batch.OfferSelectBatchJob",
                 OfferSelectBatchJob.description,
                 OfferSelectBatchJob.paramDescriptions,
                 OfferSelectBatchJob.paramDefaults);

batchAgent.registerDataSource("SDDS");
```

For the purposes of this example, although some of the batch jobs have redundant functional purposes (for example, `LearningBatchJob.java` and `LearningWithDirectEventRecordingBatchJob.java`), all of the .java files are registered in this Inline Service.

8.4.2 Performance Goals

Two performance goals, Customer Retention and Revenue, both maximized, are included in this Inline Service. Their associated scoring methods are configured at the Offers choice group level.

8.4.3 Decisions

Two decisions are configured for the CrossSellBatch Inline Service:

- OfferDecision - Based on the OfferAcceptance model
- RandomDecision - Produces a random offer for the control group

These decisions are called through the Offer Request advisor integration point. Table 8–2 and Table 8–3 describe the parameters for these decisions.

Table 8–2 Decision OfferDecision

Parameter	Description
Decision Name	OfferDecision
Select Choices From	Offers
Number of Choices to Select	1
Select at Random	No
Target Segments	Segment To Retain (Customer Retention - 70%, Revenue - 30%) Default (Customer Retention - 30%, Revenue - 70%)
Pre Selection Logic	None
Post Selection Logic	Logic included to update the choice model for the Delivered event, and to add the presented offer to the session attribute, Presented Offers

Table 8–3 Decision RandomDecision

Parameter	Description
Decision Name	RandomDecision
Select Choices From	Offers
Number of Choices to Select	1
Select at Random	Yes
Target Segments	None
Pre Selection Logic	None
Post Selection Logic	Logic included to update the choice model for the Delivered event, and to add the presented offer to the session attribute, Presented Offers

8.4.4 Choice Groups and Choices

The batch use cases for the CrossSellBatch Inline Service are demonstrated through a cross sell example. The choice groups configured represent a sample choice group hierarchy for cross selling credit card and loan offers for a fictitious financial institution. The resulting hierarchy of choice groups is depicted in the following diagram, where each of their attributes is inherited from the parent choice group, Offers.

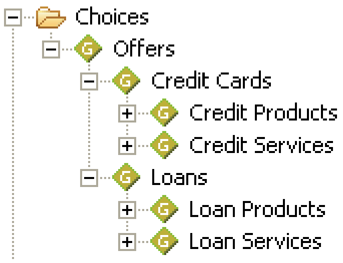


Table 8–4 describes the configuration parameters for the Offers choice group, including a list of the choice attributes.

Table 8–5 describes the choice attributes in more detail.

Table 8–4 Configuration Parameters for Offers Choice Group

Parameter	Description
Choice Group Name	Offers
Choice Attributes	Likelihood of Purchase Message Profit Margin Should Respond Positively For choice attribute details, see Table 8–5.
Scores	For the performance goal Customer Retention, score is determined at the individual choice group level. For the performance goal Revenue, score is determined by the multiplication of the value for Profit Margin times the value predicted by OfferAcceptance model for the Purchased event.
Choice Events	Delivered Interested Purchased
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 8–5 Choice Attribute Details for Offers Choice Group

Choice Attribute	Type	Value
Likelihood of Purchase	Double	Predicted by OfferAcceptance model for the Purchased event
Message	String	Defined at each individual choice
Profit Margin	Double	0.5 (value changed at the individual choice level)
Should Respond Positively	Boolean	Determined through the ShouldRespondPositively function with the input "this" choice

8.4.5 Entities

This section outlines the entity structure used for the CrossSellBatch Inline Service. As this covers a basic cross sell use case for a given set of customers, the session entity structure is kept at a minimum.

For this example the following entities have been configured:

- Customer - Stores customer data used during batch learning and scoring simulations executed through the batch framework

8.4.5.1 Session Entity

The Session entity contains all of the additional entities and attributes to be used for learning and prediction by the Inline Service.

Table 8–6 Session Entity

Attribute	Type	Array	Mapping	Default Value	Comments
Channel Id	Integer	No	None	None	Session Key
Channel	String	No	None	"Call"	None
Customer	Customer	No	None	None	None
Customer Preferences	String	Yes	CustomerPreferencesDataSource.Response	None	None
Presented Offers	Choice	Yes	None	None	None

8.4.5.2 Customer Entity

The Customer entity is associated with the session through the Session entity.

Table 8–7 Customer Entity

Attribute	Type	Array	Mapping	Default Value	Comments
Customer Id	Integer	No	None	Session.CustomerID	None
Age	Integer	No	CustomerDataSource.Age	35	None
Amount of Pending Transactions	Double	No	CustomerDataSource.AmountOfPendingTransactions	500	None
Available Credit As Percent of Credit Line	Double	No	CustomerDataSource.AvailableCreditAsPercentofCreditLine	50	None
Call Reason	String	No	CustomerDataSource.CallReason	"Credit limit related questions"	None
Calls Abandoned	Integer	No	CustomerDataSource.CallsAbandoned	1	None
Calls Last 6 Months	Integer	No	CustomerDataSource.CallsLast6Months	2	None
Card Type	String	No	CustomerDataSource.CardType	"Gold"	None
Complaints Per Year	Integer	No	CustomerDataSource.ComplaintsPerYear	1	None
Credit Line Amount	Integer	No	CustomerDataSource.CreditLineAmount	1000	None

Table 8–7 (Cont.) Customer Entity

Attribute	Type	Array	Mapping	Default Value	Comments
Day Of Week	String	No	CustomerDataSource.DayOfWeek	"Wednesday"	None
Days To Due Date	Integer	No	CustomerDataSource.DaysToDueDate	15	None
Has Credit Protection	String	No	CustomerDataSource.HasCreditProtection	"NO"	None
Language	String	No	CustomerDataSource.Language	"English"	None
Last Statement Balance	Double	No	CustomerDataSource.LastStatementBalance	400	None
Marital Status	String	No	CustomerDataSource.MaritalStatus	"Single"	None
Minimum Amount Due	Double	No	CustomerDataSource.MinimumAmountDue	50	None
Number Of Children	Integer	No	CustomerDataSource.NumberOfChildren	2	None
Occupation	String	No	CustomerDataSource.Occupation	"Managerial"	None
Signed Up For EPay	String	No	CustomerDataSource.SignedUpForEPay	"No"	None
Tenure	Double	No	CustomerDataSource.Tenure	6	None

8.4.6 Informants

The included informants for this Inline Service are the same as found in the Cross Sell example Inline Service provided with the Oracle RTD platform. As with the Cross Sell Inline Service, they are used by the batch processes to simulate learnings and provide offer predictions and scores.

The following informants are configured in the CrossSellBatch Inline Service:

- Call Info
- Call Resolution
- Call Start
- Delete All Operational Data
- Force Learning
- Offer Response
- Record Events

8.4.6.1 Call Info

The Call Info informant provides Oracle RTD with additional information about the call.

Table 8–8 Informant Call Info

Parameter	Description
Informant Name	Call Info
Session Keys	Session / Customer Id
Request Data	Channel (String) mapped to session.channel

Table 8–8 (Cont.) Informant Call Info

Parameter	Description
External System	CRM
Order	2
Force session close	No
Logic	None
Pre-condition	None

8.4.6.2 Call Resolution

The Call Resolution informant sends the final outcome of the customer call.

Table 8–9 Informant Call Resolution

Parameter	Description
Informant Name	Call Resolution
Session Keys	Session / Customer Id
Request Data	Channel (String) mapped to session.channel
External System	Fulfillment
Order	5
Force session close	Yes
Logic	None
Pre-condition	None

8.4.6.3 Call Start

The Call Start informant initiates the Oracle RTD session for the customer call.

Table 8–10 Informant Call Start

Parameter	Description
Informant Name	Call Start
Session Keys	Session / Customer Id
Request Data	Channel (String) mapped to session.channel
External System	IVR
Order	1
Force session close	No
Logic	Logic is included to fill the customer entity given the provided customer Id.
Pre-condition	None

8.4.6.4 Delete All Operational Data

The Delete All Operational Data informant is created for this example to provide a method through the application to clear out the Oracle RTD learning models and choice history between each batch simulation. Typically, this informant would not be configured in a business workflow as Delete All Operational Data is also available as a method through the platform's JMX console.

Table 8–11 Informant Delete All Operational Data

Parameter	Description
Informant Name	Delete All Operational Data
Session Keys	None
Request Data	None
External System	None
Order	0
Force session close	No
Logic	Contains logic that removes the studies associated with this Inline Service, and deletes any data from the statistics and choice event history table.
Pre-condition	None

8.4.6.5 Force Learning

The Force Learning informant was created for this example to run before batch learning. It wakes up the learning server or learning server will take about 15 seconds to wake up. This is important to do it before batch prediction, not regular real time predictions.

Table 8–12 Informant Force Learning

Parameter	Description
Informant Name	Force Learning
Session Keys	None
Request Data	None
External System	None
Order	0
Force session close	Yes
Logic	Contains logic that wakes up the Oracle RTD learning server and flushes out its buffer.
Pre-condition	None

8.4.6.6 Offer Response

The Offer Response informant is created for this example to simulate offer responses. It will create a "response" based on the CustomerPreference data source.

Table 8–13 Informant Offer Response

Parameter	Description
Informant Name	Offer Response
Session Keys	Session / Customer Id
Request Data	Channel (String) mapped to session.channel Choice Name (String) Choice Outcome (String) Simulation Mode (String)
External System	CRM

Table 8–13 (Cont.) Informant Offer Response

Parameter	Description
Order	4
Force session close	Yes
Logic	Contains the simulation logic that sets whether the customer "accepts" the offer presented.
Pre-condition	None

8.4.6.7 Record Events

The Record Events informant is created for this example to simulate an offer response. It records the event that is passed through to it as a request input parameter for the given choice.

Table 8–14 Informant Record Events

Parameter	Description
Informant Name	Record Events
Session Keys	Session / Customer Id
Request Data	Channel (String) mapped to session.channel Choice Name (String) Choice Outcome (String) Event Time (String)
External System	None
Order	0
Force session close	No
Logic	Contains the logic that sets records the event for the given outcome through the request input parameters.
Pre-condition	None

8.4.7 Advisors

The included advisors for this Inline Service are the same as found in the Cross Sell example Inline Service provided with the Oracle RTD platform. As with the Cross Sell Inline Service, they are used by the batch processes to simulate learnings and provide offer predictions and scores.

The following advisor is configured in the CrossSellBatch Inline Service:

- Offer Request

8.4.7.1 Offer Request

The advisor Offer Request returns the top ranked offers for the given customer ID.

Table 8–15 Advisor Offer Request

Parameter	Description
Advisor Name	Offer Request
Session Keys	Session / Customer Id
Request Data	Channel (String) mapped to session.channel

Table 8–15 (Cont.) Advisor Offer Request

Parameter	Description
External System	CRM
Order	3
Force session close	No
Decision	OfferDecision
Group Decision	RandomDecision
Default Choices	None
Logic	None
Pre-condition	None

8.4.8 Models

One model is included in the CrossSellBatch Inline Service. This model is used to learn on the positive acceptance events that are simulated for the presenting of each offer for each customer session.

Table 8–16 OfferAcceptance Event Model

Parameter	Description
Model Name	OfferAcceptance
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Offers
Base Event	Presented
Positive Outcome Events	Interested Purchased
Partitioning Attributes	Session / Channel
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	Keep for 30 Days using session key, customer ID

8.4.9 Functions

This section catalogs the functions included in the CrossSellBatch Inline Service, which are used for determining profit margin and for simulating customer responses for the batch job.

Table 8–17 CrossSellBatch Inline Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Multiply	A (Double) B (Double)	Double	Various objects	This function multiplies the given numbers.
ShouldRespondPositively	Choice (Choice)	Boolean	OfferResponse integration point, Offers choice attribute	Returns either True or False based on the customer preference settings in the session attribute of the same name.

8.5 Running Batch Processing

After the CrossSellBatch Inline Service is deployed, the configured batch jobs should be run through Oracle RTD's batch framework. For further information on using the batch framework, refer to the Oracle RTD Batch Framework chapter in *Oracle Real-Time Decisions Platform Developer's Guide*.

From a workflow perspective, the java batch files should be run in the following sequence:

1. **OfferSelectBatchJob.java**

   This batch process reads the `CrossSellBatchCustomers` table and writes the output to the `CrossSellBatchBestOffer` table.

2. Either of the following:

   a. **OfferAcceptanceETLBatchJob.java**

      This batch process reads from the `CrossSellBatchBestOffer` table and writes to the `CrossSellBatchLearnings` table.

   b. **OfferAcceptanceETLTableInserterBatchJob.java**

      This batch process reads from the `CrossSellBatchBestOffer` table and writes to the `CrossSellBatchLearnings` table.

3. Either of the following:

   a. **LearningBatchJob.java**

      This batch process reads the `CrossSellBatchLearnings` table.

   b. **LearningWithDirectEventRecordingBatchJob.java**

      This batch process reads the `CrossSellBatchLearnings` table.

Although they can be run independently, the Offer Scoring batch jobs (**OfferScoringBatchJob.java** and **OfferScoringBatchJobReadahead.java**) should ideally be run after 3a or 3b in order to produce scores which use the data learned in Oracle RTD from the learning batches.

Oracle RTD Decision Management Application Metadata and Inline Services

Part 4 describes the component elements of the released Inline Services and Decision Management metadata for Oracle RTD Decision Management applications.

Oracle RTD Decision Management provides a general-purpose application development tool, that enables you to build and use Decision Management applications for any business area.

Oracle RTD for Marketing Optimization application (also referred to as the RTD for Marketing Optimization application) is released with Oracle RTD Base Application as a sample marketing application. You can either use RTD for Marketing Optimization as released or adapt it to suit your own marketing requirements.

RTD for Marketing Optimization, which includes the Inline Service RTD_Base_Marketing and associated Decision Management metadata, also serves as an example for any Decision Management application in any organization.

An additional version of the Oracle RTD for Marketing Optimization application is also included with the addition of the Slices feature. For more information on this application, refer to Section 2.3, "Configuring and Deploying Base Marketing with Slices" in *Oracle Real-Time Decisions Base Application Decision Management Installation and Configuration Guide*.

You can build your own Decision Management application by starting with a basic template Inline Service, RTD_CLM_Core, and its associated "core" Decision Management metadata, which are also released with Oracle RTD Base Application. Together these components do not constitute a complete Decision Management application, but they contain the starting-point building blocks for any Decision Management application. You can add and configure the elements specific to your organization by examining corresponding or similar elements in RTD for Marketing Optimization, and then adapting them to your particular requirements.

Part 4 contains the following chapters:

- Chapter 9, "Elements in Core and Base Marketing Inline Services"
- Chapter 10, "Decision Management Setup Files"

Elements in Core and Base Marketing Inline Services

Terminology: The following terms are used throughout this chapter:

- **Core ILS** *(or **Core Inline Service**)* - refers to the RTD_CLM_Core Inline Service that serves as the basis for all Oracle RTD Decision Management applications
- **Base Marketing ILS** *(or **Base Marketing Inline Service**)* - refers to the RTD_Base_Marketing Inline Service released with Oracle RTD Base Application as part of the RTD for Marketing Optimization application

---

This chapter focuses on the Inline Service elements required for Oracle RTD Decision Management applications, as they appear in the Core ILS and the Base Marketing ILS.

Note: The Base Marketing Slices ILS used by the Base Marketing with Slices application is the same as the Base Marketing ILS. For more information on the Base Marketing with Slices application, see Section 2.3 "Configuring and Deploying Base Marketing with Slices" in the *Oracle Real-Time Decisions Base Application Decision Management Installation and Configuration Guide*.

This chapter also references the metadata required as additional components for Oracle RTD Decision Management applications. For details of these metadata elements, see the following:

- Chapter 10, "Decision Management Setup Files"

- The chapter "Configuring Oracle Decision Management" in *Oracle Real-Time Decisions Base Application Decision Management Installation and Configuration Guide*

This section contains the following topics:

- Section 9.1, "Overview of Core and Base Marketing Inline Service Elements"
- Section 9.2, "Decision Management Inline Service Elements"

9.1 Overview of Core and Base Marketing Inline Service Elements

Table 9–1 shows a summary of the basic elements related to all Oracle RTD Decision Management Inline Services (in the Core ILS column) and those elements specific to the Inline Service for the RTD for Marketing Optimization application (in the Base Marketing ILS column).

Table 9–1 Elements in Core and Base Marketing Inline Services for Oracle RTD Decision Management Applications

ILS Elements	Core ILS	Core Ref	Base Marketing ILS	Base Mktg Ref
Application Logic	Initialization + Cleanup logic	"Application"	As in Core ILS	NA
Application parameters	CLM Database Polling Delay In Seconds CLM ILS Choice Groups CLM JDBC Source	"Application" "Dynamic Choice Cache Administration Methods"	As in Core ILS + ▪ Record Decision Event Log ▪ Record Choice Event Log	NA
Session entity	Attribute: CLM Version Initialization + Cleanup logic	"Session"	As in Core ILS	NA
Entities	CLM Dummy Entity	"Choice Groups" "Entities"	As in Core ILS + ▪ Customer	"Entities"
Data Sources	*(None)*	NA	CustomerDataSource CustomerPreferencesDataSource	NA
Choice Groups	CLM Base	"Choice Groups" "Propagation of Rules"	As in Core ILS + ▪ Campaign ▪ Channel ▪ Creative ▪ Decision ▪ Offer ▪ Placement ▪ Project ▪ Segment ▪ Slot ▪ Slot Type ▪ Tag *(All these extra choice groups are under the CLM Base choice group)*	"Choice Groups"
Decisions	*(None)*	NA	Creative Decision Random Creative Decision	"Integration Points" "Decisions" "Architecture Overview"

Table 9–1 (Cont.) Elements in Core and Base Marketing Inline Services for Oracle RTD Decision Management Applications

ILS Elements	Core ILS	Core Ref	Base Marketing ILS	Base Mktg Ref
Models	*Statistics (general model, not oriented to Decision Management)*	NA	Campaign Acceptance Channel Acceptance Creative Acceptance Offer Acceptance Placement Acceptance Slot Acceptance Slot Type Acceptance Tag Acceptance	"Modeling"
Functions	CLM Is Choice Graph Eligible CLM Is Choice Eligible CLM Record Event Including Related Choices	"Propagation of Rules" "Propagation of Events"	As in Core ILS+ ▪ CLM Get Weights ▪ CLM Record Log ▪ Get CLM ILS Channel Creatives ▪ Get CLM ILS Campaign Offers ▪ Get CLM ILS Channel Placements ▪ Get CLM ILS Creative Channel ▪ Get CLM ILS Creative Offer ▪ Get CLM ILS Creative Slot Type ▪ Get CLM ILS Creative Tags ▪ Get CLM ILS Offer Campaign ▪ Get CLM ILS Offer Creatives ▪ Get CLM ILS Placement Channel ▪ Get CLM ILS Placement Slots ▪ Get CLM ILS Slot Placement ▪ Get CLM ILS Slot Slot Type ▪ Get CLM ILS Slot Type Creatives ▪ Get CLM ILS Slot Type Slots ▪ Get CLM ILS Tag Creatives ▪ Get CLM Marketing Priority ▪ GetDecision	"Choice Groups" "Decisions" "External Performance Goal Editor" "Offer Choice Group" "Decision Analytics Logging"
Informants	CLM Request Database Hard Refresh CLM Request Database Soft Refresh CLM Reset Project Cache	"Dynamic Choice Cache Administration Methods"	As in Core ILS + ▪ Creative Feedback ▪ Session Resolution ▪ Session Start	"Integration Points" "Loading the Choices of Active Projects in an Inline Service"

Table 9–1 (Cont.) Elements in Core and Base Marketing Inline Services for Oracle RTD Decision Management Applications

ILS Elements	Core ILS	Core Ref	Base Marketing ILS	Base Mktg Ref
Advisors	*(None)*	NA	Get Creative	"Integration Points"
Type Restrictions	*(None)*	NA	Approval Status Eligibility Rule Filtering Rule Project Type Region Scroring Rule Type	"Type Restrictions"

Java classes (in `src/com/sigmadynamics/sdo`)	`ApplicationSession.java`
	`CLMChoiceBag.java`
	`CLMChoiceBagStore.java`
	`CLMChoiceData.java`
	`CLMChoiceGroupMapper.java`
	`CLMChoiceGroupMapperStore.java`
	`CLMChoiceInstanceStore.java`
	`CLMCustomHandler.java`
	`CLMCustomHandlerStore.java`
	`CLMData.java`
	`CLMDatabaseHelper.java`
	`CLMDatabaseLoader.java`
	`CLMDataStore.java`
	`CLMMetadata.java`
	`CLMMetadataStore.java`
	`CLMNotRunningException.java`
	`CLMPropagateEnum.java`
	`CLMRelationshipKey.java`
	`CLMRelationshipType.java`
	`CLMScheduledExecutorService.java`
	`CLMShutdownException.java`
		"Java APIs"	As in Core ILS +
			▪ `CreativeBySlotTypeHandler.java`
			▪ `CLMPerformanceGoalWeights.java`
			▪ `CLMPerformanceGoalWeightsEsception.java`
			▪ `CLMPerformanceGoalWeightsHandler.java`
				NA

9.2 Decision Management Inline Service Elements

This section describes the elements that occur in Inline Services that form part of a Oracle RTD Decision Management application. Some elements are general, that is, must exist in all such Inline Services, others are particular to the Base Marketing Inline Service associated with the reference application, RTD for Marketing Optimization, released with Oracle RTD Base Application.

As in other sections of this chapter, the terms **Core ILS** and **Base Marketing ILS** are used as required to specify where the elements are defined. Table 9–1 provides a summary of which elements are common to both Inline Services and which are specific to the Base Marketing ILS.

This section contains the following topics:

- Section 9.2.1, "Architecture Overview"
- Section 9.2.2, "Application"

- Section 9.2.3, "Session"
- Section 9.2.4, "Entities"
- Section 9.2.5, "Choice Groups"
- Section 9.2.6, "Modeling"
- Section 9.2.7, "Integration Points"
- Section 9.2.8, "Decisions"
- Section 9.2.9, "Type Restrictions"
- Section 9.2.10, "Propagation of Rules"
- Section 9.2.11, "Propagation of Events"
- Section 9.2.12, "External Performance Goal Editor"
- Section 9.2.13, "Decision Analytics Logging"
- Section 9.2.14, "Java APIs"

9.2.1 Architecture Overview

All Inline Services that form part of a Oracle RTD Decision Management application share a common architectural structure.

The Base Marketing ILS is a sample Inline Service that shows how to use choices managed by a Oracle RTD Decision Management application. This sample Inline Service uses the choice groups and relationship types of the RTD for Marketing Optimization application.

The Core ILS is a basic Inline Service which can be added to, to create an Inline Service specific to other Oracle RTD Decision Management applications.

[When you build your own Oracle RTD Decision Management application, you have to add application-specific elements and steps in addition to those provided with the Core ILS. **You are strongly advised to refer to the application-specific elements and steps in the Base Marketing ILS to see the corresponding elements and steps there.**]

In all Oracle RTD Decision Management Inline Services, you need to define the same choice groups in the Inline Service as you define in the Oracle RTD Decision Management metadata. In addition, you can define choice attributes that follow relationship types between choice groups using the sample functions provided in the Base Marketing ILS.

Oracle RTD Decision Management Inline Services use their own mechanism to load dynamic choices, instead of using an entity as explained in *Oracle Real-Time Decisions Developer's Guide*. The settings explained in *Oracle Real-Time Decisions Developer's Guide* are mandatory for dynamic choice groups, therefore "dummy" objects have been introduced for that purpose, but are not used. The load of dynamic choice groups happens automatically. The only difference to note is that the choice array passed as an argument to decisions pre-selection logic is empty. The **Creative Decision** decision in the Base Marketing ILS shows an example of how to properly handle this.

These design decisions enable the following application design goals:

- Easier setup of the Inline Service

  You do not have to specify a data source and an entity for each dynamic choice group and wire all these objects manually. This makes creation and maintenance of the Inline Service much easier.

- Better control of dynamic choices lifecycle

  The Base Marketing Inline Service automatically detects when dynamic choices have been updated in the Decision Management database main repository. This is done asynchronously and therefore has no impact on integration point response time. Dynamic choices are loaded asynchronously and their rules are compiled asynchronously. Integration point requests for new sessions will start using a new version after this whole asynchronous process has completed. To keep cohesion within a session, multiple integration point requests in the same sessions all use the same version of dynamic choices.

- Ability to retrieve custom lists of dynamic choices

  During the asynchronous load of the choices from the Oracle RTD Decision Management database, you can mark choices as belonging to any list you want. During an integration point, you can retrieve the dynamic choices of that list in a single call, which saves having to do the same time consuming operations over and over in each integration point.

- Ability to run tests against a project before it is committed to the main repository

The Oracle RTD Decision Management database main repository gets updated to a new version when either of the following occurs:

- Someone commits a project using the Decision Manager user interface on the same database
- Changes from a different system are pushed to this production Oracle RTD Decision Management database using an offline process

9.2.2 Application

The application has the following parameters, common to all Oracle RTD Decision Management Inline Services:

- **CLM Database Polling Delay In Seconds**: the number of seconds the Inline Service will wait between the beginning of the previous poll and a new poll, to see if there is a new repository version in the Oracle RTD Decision Management database
- **CLM ILS Choice Groups**: an array of all the dynamic choice groups to be loaded from the Oracle RTD Decision Management database
- **CLM JDBC Source**: the datasource you have added in Oracle RTD's web.xml for rtis.war, typically CLMDS

The application object has some code in initialization logic and cleanup logic to start and stop the asynchronous load of Oracle RTD Decision Management dynamic choices.

For the Base Marketing ILS, the following two parameters are added:

- **Record Decision Event Log**: specifies if logging of decisions is done
- **Record Choice Event Log**: specifies if logging of choice events is done

9.2.3 Session

Common to all Oracle RTD Decision Management Inline Services, the Session entity has an attribute called **CLM Version**, which is used to track which repository version of the Oracle RTD Decision Management database this session is using. If a project is committed, current sessions will continue to use the previous repository version of the

Oracle RTD Decision Management database, and new sessions will start using the new repository version of the Oracle RTD Decision Management database after it is loaded in Inline Service memory.

The session has some lines of code in the Initialization logic and Cleanup logic that should not be removed.

9.2.4 Entities

CLM Dummy Entity appears in both the Core ILS and the Base Marketing ILS, and is required to satisfy dynamic choice group definitions. The Customer entity appears in the Base Marketing ILS only.

9.2.4.1 CLM Dummy Entity

The CLM Dummy Entity entity has the following attributes:

- Dummy Choice ID

9.2.4.2 Customer (in Base Marketing ILS only)

The Customer entity has the following attributes:

- Customer ID
- Age
- Amount Of Pending Transactions
- AvailableCreditAsPercentOfCreditLine
- CallReason
- CallsAbandoned
- CallsLast6Months
- CardType
- ComplaintsPerYear
- CreditLineAmount
- DayOfWeek
- Days To Due Date
- HasCreditProtection
- Language
- LastStatementBalance
- Marital Status
- MinimumAmountDue
- NumberOfChildren
- Occupation
- Signed Up For Epay
- Tenure

9.2.5 Choice Groups

The dynamic choices must be created under the CLM Base choice group.

Note: The CLM Base choice group exists in both the Core ILS and the Base Marketing ILS.

In the Core ILS, there are no choice groups defined under CLM Base. The Base Marketing ILS contains the choice groups described later in this section.

For all Oracle RTD Decision Management application Inline Services, in the Dynamic Choices tab of each choice group under CLM Base, select the option Use Dynamic Choices for this Choice Group, and specify these properties:

- Group attribute containing the list of Entities for choices: CLM Dummy Entity List
- Choice attribute to assign the entity data: Dummy Data
- Entity attribute that contains the choice id: Dummy Choice Id

You must create choice groups and choice attributes with ids, data types and type restrictions that match the ones specified in the Oracle RTD Decision Management metadata configuration files. Only Boolean, Date, Double, Integer and String data types are supported. Arrays are not supported.

The choice groups defined in the Base Marketing ILS are as follows:

- Campaign Choice Group
- Offer Choice Group
- Creative Choice Group
- Tag Choice Group
- Channel Choice Group
- Placement Choice Group
- Slot Choice Group
- Slot Type Choice Group
- Project Choice Group
- Segment Choice Group
- Decision Choice Group

Project Choice Group

The Project choice group is required in the Inline Service if the Decision Management application has project attributes with type restrictions. Type restrictions provide lists of values for the attributes during project creation and editing.

Note: The Project choice group must be defined in the metadata for a Decision Management application, even if no project attributes have type restrictions.

Functions for Choice Group Relationship Attributes

You must create functions to create the relationship attributes between choices. See the following examples in the Oracle RTD Base Marketing Inline Service:

Note: In some functions, the relationship returns one choice (such as between one offer and its campaign); in other functions, the relationship returns multiple choices (such as between one campaign and its multiple offers).

- Get CLM ILS Campaign Offers
- Get CLM ILS Channel Creatives
- Get CLM ILS Channel Placements
- Get CLM ILS Creative Channel
- Get CLM ILS Creative Offer
- Get CLM ILS Creative Slot Type
- Get CLM ILS Creative Tags
- Get CLM ILS Offer Campaign
- Get CLM ILS Offer Creatives
- Get CLM ILS Placement Channel
- Get CLM ILS Placement Slots
- Get CLM ILS Slot Placement
- Get CLM ILS Slot Slot Type
- Get CLM ILS Slot Type Creatives
- Get CLM ILS Slot Type Slots
- Get CLM ILS Tag Creatives

Note: in the following tables for each of the choice group attributes:

- DM refers to the Decision Manager user interface available with Oracle RTD Decision Management
- DC refers to the Decision Center user interface available with Oracle RTD

9.2.5.1 Campaign Choice Group

Table 9–2 shows the non-relationship attributes for the Campaign choice group.

Table 9–2 Campaign Choice Group Attributes

Attribute Name	Data Type	DM Visible?	Required?	Type Restricted?	DC Viewable?
Approval Status	String (20)	Yes	Yes	Yes	Yes
Code	String (20)	Yes	Yes	No	Yes
Region	String (20)	Yes	No	Yes	Yes
Start Date	Date Time	Yes	No	No	Yes
End Date	Date Time	Yes	No	No	Yes
Type	String (20)	Yes	Yes	Yes	Yes

9.2.5.2 Offer Choice Group

Table 9–3 shows the non-relationship attributes for the Offer choice group.

Table 9–3 Offer Choice Group Attributes

Attribute Name	Data Type	DM Visible?	Required?	Type Restricted?	DC Viewable?
Code	String (20)	Yes	Yes	No	Yes
Type	String (20)	Yes	Yes	Yes	Yes
Start Date	Date Time	Yes	No	No	Yes
End Date	Date Time	Yes	No	No	Yes
Approval Status	String (20)	Yes	Yes	Yes	Yes
Region	String (20)	Yes	No	Yes	Yes
Product	String (40)	Yes	No	No	Yes
Promotion	String (40)	Yes	No	No	Yes
Cost	Double	Yes	No	No	Yes
Revenue	Double	Yes	No	No	Yes
Marketing Priority	String (CLOB)	Yes	No	No	No
Marketing Priority Value	Double	No	No	No	No

The *Marketing Priority Value* is used to cache the value returned by the Marketing Priority scoring rule for the duration of the integration point. It is computed by calling the `Get CLM Marketing Priority` function.

9.2.5.3 Creative Choice Group

Table 9–4 shows the non-relationship attributes for the Creative choice group.

Table 9–4 Creative Choice Group Attributes

Attribute Name	Data Type	DM Visible?	Required?	Type Restricted?	DC Viewable?
Approval Status	String (20)	Yes	Yes	Yes	Yes
Code	String (20)	Yes	Yes	No	Yes
Start Date	Date Time	Yes	No	No	Yes
End Date	Date Time	Yes	No	No	Yes
Type	String (20)	Yes	Yes	Yes	Yes
Region	String (20)	Yes	No	Yes	Yes
Product	String (40)	Yes	No	No	Yes
Promotion	String (40)	Yes	No	No	Yes
Cost	Double	Yes	No	No	Yes
Revenue	Double	Yes	No	No	Yes
Image	String (250)	Yes	No	No	Yes
Script	String *(CLOB)*	Yes	No	No	Yes

9.2.5.4 Tag Choice Group

The Tag choice group contains only relationship attributes, defined in the Tag-related Functions for Choice Group Relationship Attributes.

Note: in the Inline Service, Tag choice groups are defined with default eligibility rules. However, unlike the other choice groups, there are no equivalent eligibility rules for Tag choices in Decision Management metadata.

9.2.5.5 Channel Choice Group

The Channel choice group contains only relationship attributes, defined in the Channel-related Functions for Choice Group Relationship Attributes.

9.2.5.6 Placement Choice Group

Table 9–5 shows the non-relationship attributes for the Placement choice group.

Table 9–5 Placement Choice Group Attributes

Attribute Name	Data Type	DM Visible?	Required?	Type Restricted?	DC Viewable?
Type	String (40)	Yes	No	No	Yes
URL	String (250)	Yes	No	No	Yes

9.2.5.7 Slot Choice Group

The Slot choice group contains only relationship attributes, defined in the Slot-related Functions for Choice Group Relationship Attributes (that is, functions related specifically to Slot, rather than Slot Type).

9.2.5.8 Slot Type Choice Group

The Slot Type choice group contains only relationship attributes, defined in the Slot Type-related Functions for Choice Group Relationship Attributes (that is, functions related specifically to Slot Type, rather than Slot).

9.2.5.9 Project Choice Group

The Project choice group is only required in the Inline Service if you want to see lists of values for project attributes in Decision Management applications when creating and editing projects. If no Project choice group exists in the Inline Service, Decision Management Project screens will show Name and Description only.

Table 9–6 shows the non-relationship attributes for the Project choice group.

Table 9–6 Project Choice Group Attributes

Attribute Name	Data Type	DM Visible?	Required?	Type Restricted?	DC Viewable?
Type	String (20)	Yes	Yes	Yes	Yes

9.2.5.10 Segment Choice Group

Table 9–7 shows the non-relationship attributes for the Segment choice group.

Table 9–7 Segment Choice Group Attributes

Attribute Name	Data Type	DM Visible?	Required?	Type Restricted?	DC Viewable?
Filtering Rule	String (CLOB)	Yes	No	No	No

9.2.5.11 Decision Choice Group

Table 9–8 shows the non-relationship attributes for the Decision choice group.

Table 9–8 Decision Choice Group Attributes

Attribute Name	Data Type	DM Visible?	Required	Type restricted?	DC Viewable?
Weights	String (CLOB)	Yes	No	No	No

9.2.6 Modeling

There are eight choice event models in the Base Marketing ILS, one for each choice group. All the choice event models are used for tracking and reports, and the **Creative Acceptance** model is also used to predict the most suitable creatives.

The Inline Service can be modified to implement more advanced decision logic involving multiple predictive models, for example, to select the best offer according to certain predictive criteria, then to pick the most suitable creative associated with this offer.

Apart from the general **Statistics** model, there is no model defined in the Core ILS.

9.2.7 Integration Points

All the integration points and decisions described in this section occur in the Base Marketing ILS only.

The **Session Start** and **Session Resolution** informants are typical informants for opening and closing sessions.

The **Get Creative** advisor returns a creative for a given slot. A creative has a slot type and a slot has a slot type. Only creatives that have a slot type similar to the slot type of the slot passed as incoming parameter will be returned. This advisor uses the **Creative Decision** decision and the **Random Creative Decision** decision for the control group.

The **Creative Feedback** informant closes the loop by recording the outcome for that creative and that slot

See also Section 9.2.14.6, "Loading the Choices of Active Projects in an Inline Service" and Section 9.2.14.7, "Dynamic Choice Cache Administration Methods" for information about project administration informants.

9.2.8 Decisions

All the decisions described in this section occur in the Base Marketing ILS only.

The **Creative Decision** decision and the **Random Creative Decision** decision use the same code in pre-selection logic and post-selection logic to return only creatives that are eligible and have the same slot type as the slot passed as argument. The logic records that this creative was presented for both that creative and that slot.

The weights for the decision are defined in Decision Manager and retrieved using the Decision choice with the id specified as argument in the custom goal weights function, which is the following by default: `CLM Get Weights("CreativeDecision")`.

9.2.9 Type Restrictions

All the type restrictions, choice groups, and choice group attributes described in this section occur in the Base Marketing ILS only.

In the Base Marketing Inline Service, type restrictions are defined for the attributes of certain choice groups. This section lists the type restrictions and the choice group attributes that use them.

This section contains the following topics:

- Approval Status Type Restriction
- Project Type Type Restriction
- Region Type Restriction
- Type Type Restriction
- Eligibility Rule, Filtering Rule and Scoring Rule Type Restrictions
- Choice Group Attribute Usage of the Type Restrictions

9.2.9.1 Approval Status Type Restriction

Table 9–9 show the values for the Approval Status type restriction.

Table 9–9 Approval Status Type Restriction Values

Approval Status Values
Draft
Pending Approval
Approved

9.2.9.2 Project Type Type Restriction

Table 9–10 show the values for the Project Type type restriction.

Table 9–10 Project Type Restriction Values

Project Type Values
Emergency Changes
Incremental Changes
Structural Changes

9.2.9.3 Region Type Restriction

Table 9–11 show the values for the Region type restriction.

Table 9–11 Region Type Restriction Values

Region Values
All
NA

Table 9–11 (Cont.) Region Type Restriction Values

Region Values
LA
EMEA
APAC

9.2.9.4 Type Type Restriction

Table 9–12 show the values for the Type type restriction.

Table 9–12 Type Type Restriction Values

Type Values
Cross-Sell
Up-Sell
Winback
Retain
Acquisition

9.2.9.5 Eligibility Rule, Filtering Rule and Scoring Rule Type Restrictions

These type restrictions are used to specify which attributes should be automatically compiled as either an eligibility, filtering or scoring rule. Specify these type restrictions for attributes that use the rule editor in Decision Manager.

9.2.9.6 Choice Group Attribute Usage of the Type Restrictions

Table 9–13 shows the choice group attributes that use the type restrictions defined in the Base Marketing Inline Service.

Table 9–13 Choice Group Attribute Usage of Type Restrictions

Type Restriction	Used By
Approval Status	Campaign - Approval Status Offer - Approval Status Creative - Approval Status
Project Type	Project - Type
Region	Campaign - Region Offer - Region Creative - Region
Type	Campaign - Type Offer - Type Creative - Type
Eligibility Rule	CLM Base - Eligibility Rule
Filtering Rule	Segment - Filtering Rule
Scoring Rule	Offer - Marketing Priority

9.2.10 Propagation of Rules

The dynamic choice groups all inherit from CLM Base. Therefore they only inherit rules from CLM Base, there is no inheritance of rules from one choice group to the other as with standard, non-Decision Management Inline Service designs.

Instead, rules are propagated based on metadata defined in relationship types. In the reference implementation (RTD for Marketing Optimization), all relationship types are defined to propagate rules from destination to owner - this applies to all the choice groups except Tag, as tags do not have rules defined in the metadata. Relationship types are defined as either "many to one" or "many to zero" (where the owner is the "many" side and the destination is the "one" or "zero" side of the relationship) or "many to many".

Therefore, when eligibility is evaluated on a creative, this also evaluates eligibility on that creative's offer, channel and slot type. The offer in turn evaluates eligibility on its campaign. So, in order to be eligible, the campaign, offer, channel and slot type of that creative all have to be eligible. This is more powerful than simple inheritance because multiple "directions" can be followed. This can be thought of as multiple inheritance, or more accurately composition. The rules for propagation are defined in metadata and are therefore highly customizable.

For the rules to be properly executed, the Choice Eligibility for the CLM Base choice group calls the **CLM Is Choice Graph Eligible** function which executes the eligibility rule on this choice and all the choices it needs to propagate to.

Note that there is a **Choice Is Eligible** choice attribute in CLM Base in order to cache the eligibility for the duration of the integration point.

9.2.11 Propagation of Events

Propagation of events is done in a similar fashion to the propagation of rules.

In the Base Marketing ILS, Oracle RTD gets a creative for a slot. Oracle RTD records the event for:

- The creative and its campaign, offer, tag, channel and slot type
- The slot and its placement, channel and slot type

Oracle RTD does not count the event twice on the channel and slot type, which appear in both lists. Therefore, Oracle RTD records the event once for the creative, campaign, offer, tag, channel, slot type, and placement involved related to that creative and slot.

In order to follow these relationships while recording events, you must call the **CLM Record Event Including Related Choices** function.

9.2.12 External Performance Goal Editor

Decision Manager lets business users specify performance goal weights for a choice of type Decision.

The decision that wants to use the External Performance Goal Editor needs to call the **CLM Get Weights** function with the decision choice id as argument (the same ID as the ID of the decision choice business users are editing in Decision Manager).

9.2.13 Decision Analytics Logging

The function **CLM Record Log** records information about the decision or choice event in the Decision Analytics tables. This function is executed in the post selection logic of the Get Creative advisor and in the logic of the Creative Feedback informant. When logging occurs depends on the values of the Record Decision Log and Record Choice Event Log application parameters. The function **GetDecison** is used by **CLM Record Log** to find out the name of the decision logged in the choice event log.

For more information on Decision Analytics logging, see Section 18.2.4, "Inline Service APIs" in the *Oracle Real-Time Decisions Platform Developer's Guide*.

9.2.14 Java APIs

Note: The javadoc for the Java APIs can be found in `clm\lib\clm-ils-api-javadoc.jar`.

This section contains the following topics:

- Stores
- A Note on "ids", "names", and "labels"
- Distinctions Between the "CLM" Classes
- Custom Handlers and CLMData
- CLMChoiceBag
- Loading the Choices of Active Projects in an Inline Service
- Dynamic Choice Cache Administration Methods

9.2.14.1 Stores

Several "Store" classes are used to store objects in memory for the right lifespan.

CLMMetadataStore holds CLMMetadata in memory for the life lifespan of the Inline Service. CLMMetadata consists of information on propagation of rules and events that is stored in the Oracle RTD Decision Management database.

CLMDataStore holds CLMData in memory. CLMData consists of all the dynamic choices and all the relationships between these choices. CLMDataStore holds the latest version of that CLMData, and any older CLMData still used by existing sessions.

CLMChoiceBagStore holds a CLMChoiceBag in memory for the lifespan of each integration point. CLMChoiceBag contains dynamic choice objects that can be used in your java code to work with all the dynamic choices. See the "CLM Test Informant" for an example.

CLMCustomHandlerStore holds all the custom CLMCustomHandler instances that will be called during each new Oracle RTD Decision Management database load.

CLMChoiceGroupMapperStore and **CLMChoiceInstanceStore** are internal stores used to cache objects obtained using Java reflection APIs to improve performance.

9.2.14.2 A Note on "ids", "names", and "labels"

Some methods ask for the "choice id" of a dynamic choice, which is different from the "SDOId" of the dynamic choice. The choice id is what users see and enter in Decision Manager. The SDOId is the concatenation of the group id, the $ symbol and the choice id. The attribute "Choice Id" has been added in choice group "CLM Base" so that it is possible to retrieve the choice id of a Oracle RTD Decision Management dynamic choice by accessing that attribute.

Some methods ask for the "group id" (also known as the "choice group id") of the dynamic choice. This is the SDOId of the choice group of the dynamic choice (getGroup().getSDOId()) and corresponds to the choice group id in both CLM and Inline Service metadata.

The choice name (as entered in the Decision Manager user interface) is retrieved by calling getSDOLabel() on the dynamic choice.

9.2.14.3 Distinctions Between the "CLM" Classes

CLMDataStore holds multiple CLMData, one for each in memory version of the Oracle RTD Decision Management database.

CLMChoiceData holds data information about one choice.

CLMData holds all the choices as CLMChoiceData objects.

CLMData holds all the relationships between these objects.

CLMChoiceBagStore holds multiple CLMChoiceBag, one for each in thread currently running an integration point.

CLMChoiceBag holds choices as dynamic choices, by creating them from CLMData as needed.

The dynamic choices have relationships between each other, which are computed from CLMData as needed.

Dynamic choices are only valid for the lifespan of the integration span, so each choice bag is discarded at the end of the integration point. This is why, for performance reasons, Oracle RTD creates the dynamic choices in them. This is also one of the reasons for introducing the custom handler feature. With this feature, you can get a subset of dynamic choices without having to create other dynamic choices. In the "Get Creative" example, you can get all the creative dynamic choices of a specific slot type, without having to create a dynamic choice for every single creative, and then checking the slot type of each of these creatives, doing this every time the advisor is called.

9.2.14.4 Custom Handlers and CLMData

Custom handlers let you access CLMData to build and store custom lists.

CLMData provides APIs to:

- Retrieve a CLMChoiceData choice object given its choice id
- Retrieve all choice ids for a given choice group
- Retrieve all choice ids for a given custom list
- Retrieve choice ids related to a given choice
- Add a choice id to a custom list (the goal of custom handlers)

9.2.14.5 CLMChoiceBag

CLMChoiceBag is the main class you use in your Inline Service to access dynamic choices.

It provides APIs to:

- Retrieve a dynamic choice object given its choice id
- Retrieve all dynamic choices (or their ids) for a given choice group
- Retrieve all dynamic choices (or their ids) for a given custom list

- Retrieve choices (or their ids) related to a given choice

9.2.14.6 Loading the Choices of Active Projects in an Inline Service

You can load the main repository choices and the changes that have been made in a project. This can be used to test the changes in a project before committing them. See the method CLMChoiceBagStore.setProject(int projectRowId). The choices loaded for that project are cached in memory.

9.2.14.7 Dynamic Choice Cache Administration Methods

Oracle RTD Decision Management Inline Services must check to see if changes have been made due to a project being committed at the regular interval defined in the Application parameter **CLM Database Polling Delay In Seconds**.

Two informants have been added to control that behavior:

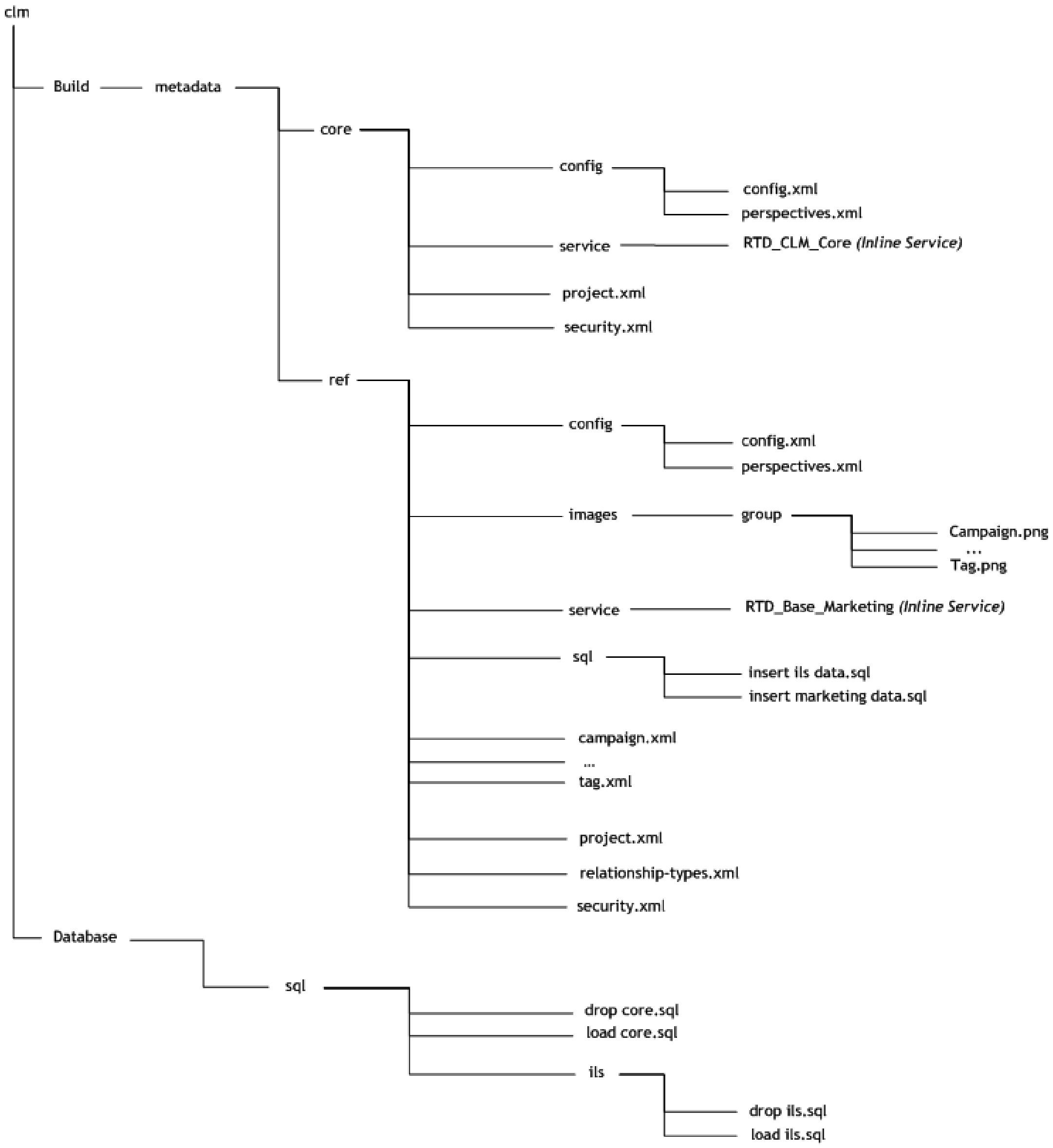
- **CLM Request Database Hard Refresh**: requests a refresh of all dynamic choices and starts the polling interval again after the refresh is done
- **CLM Request Database Soft Refresh**: requests a refresh of all dynamic choices only if a project has been committed since last refresh, and starts the polling interval again after the refresh is done

Another informant, **CLM Reset Project Cache**, shows how to reset the cache of the choices for a given project.

Decision Management Setup Files

The following diagram shows the main Decision Management metadata and database setup files, as released with Oracle RTD Base Application.

Figure 10–1 Main Oracle RTD Decision Management Metadata and Database Files



Terminology: The following terms are used throughout this chapter:

- **Core** - refers to the basic elements required for all Oracle RTD Decision Management applications
- **Base Marketing** (or **Ref**) - refers to the elements for the RTD for Marketing Optimization application

---

This chapter contains the following topics:

- Section 10.1, "Core Metadata Files"
- Section 10.2, "Base Marketing (ref) Metadata Files"
- Section 10.3, "Database Files for Core and Base Marketing"

Note: The Base Marketing with Slices application is based on the Base Marketing application and contains some additional metadata. For more information on the Base Marketing with Slices application including the additional metadata, see Section 2.3, "Configuring and Deploying Base Marketing with Slices" in the *Oracle Real-Time Decisions Base Application Decision Management Installation and Configuration Guide*.

10.1 Core Metadata Files

The Core metadata files in this section all appear under the directory 22.

This section contains the following topics:

- core/config/config.xml
- core/config/perspectives.xml
- core/project.xml
- core/security.xml

core/config/config.xml

The Core `config.xml` file contains general configuration settings:

- workbenchService: the url of the Oracle RTD server running in the same host server as Oracle RTD Decision Management.
- inlineService: the name of the Core Inline Service (RTD_CLM_Core) that is released with Oracle RTD Base Application.
- decisionCenter: the base URL to access embedded decision center reports.
- ownership-mode: special mode, which when switched on, allow users (typically administrators) to take ownership control of choices when choices are edited. Initial value after installation is "off".
- display-name: the name of the application as it appears on the Decision Manager login page and at the top of the main page.

core/config/perspectives.xml

The Core `perspectives.xml` file acts as the base file for adding your application perspectives. The Core perspective file contains no specific perspectives.

core/project.xml

The Core `project.xml` file acts as the base file for adding your application projects. It contains the create/edit/view pages for the special choice group Project, and the attributes Name and Description for the Overview page.

core/security.xml

The Core `security.xml` file contains the configuration entries for the three application roles CLMConsumer, CLMAuthor, CLMAdministrator, and the choice group, project, and perspective permissions selected for those application roles. It contains no entries for any enterprise roles nor users.

10.2 Base Marketing (ref) Metadata Files

The Base Marketing (ref) metadata files in this section all appear under the directory **clm/Build/metadata**.

This section contains the following topics:

- ref/config/config.xml
- ref/config/perspectives.xml
- The Base Marketing Choice Group xml files
- ref/project.xml
- ref/relationship-types.xml
- ref/security.xml
- ref/sql/insert ils data.sql and ref/sql/insert marketing data.sql

ref/config/config.xml

The Base Marketing (ref) `config.xml` file contains general configuration settings. For details, see Section 2.2.2.1 "Config XML File" in the *Oracle Real-Time Decisions Base Application Decision Management Installation and Configuration Guide*:

ref/config/perspectives.xml

The Base Marketing (ref) `perspectives.xml` file contains the properties for the perspectives released with the RTD for Marketing Optimization application: Campaigns, Tags, Channels, Slot Types, Draft Campaigns, Pending Approval Campaigns, and Approved Campaigns.

The Base Marketing Choice Group xml files

For each of the choice groups defined in the RTD for Marketing Optimization application - Campaign, Offer, Creative, Tag, Channel, Placement, Slot, Slot Type, Project, Segment, Decision - there is a corresponding choice group XML file in the **ref** directory.

Each choice group XML file contains the properties to define the choice group and its attributes. Other properties control whether the choice group appears in Decision Manager dropdown lists, and whether its attributes appear by default in Decision Manager Advanced Search screens. Type restriction associations are defined for several attributes in the Campaign, Offer, Creative, and Project choice groups.

Each choice group XML file also contains the create/edit/view pages that define the pages that appear when the choice group is created, edited, or viewed in the RTD for Marketing Optimization application.

The Base Marketing Choice Group XML files have been updated to add two new tabs to display Model Reports; that is, the Performance tab and the Analysis tab.

Each choice group XML file also contains tabs that define the pages that appear when the choice group viewed in the RTD for Marketing Optimization application.

The following code sample shows the metadata changes as updated to display tabs and model reports.

```
<tab id="<id>" type="<type>">
<model-report reportType="<type>"/>
```

Where `type` is: `choiceHistory`, `predReport`, `bestFit`, `drivers`, `modelQuality`.

To include a tab in a choice group, add the following code to the respective choice group's XML (such as `campaign.xml`, `channel.xml`, and so on) in the metadata folder. Each page should have one `<page id="" viewType="view/edit/create">` element. For instance, if a tab displays four pages, it should contain four page attributes within the `<tab>` element; for example:

```
<tab id="<id>" type="type">
   <name>Name</name>
   <description>Description</description>
   <page id="" viewType="view/edit/create">
      <name></name>
      <description></description>
      <page-attributes-group>
         <model-report reportType="<type>"/>
      </page-attributes-group>
   </page>
</tab>
```

If `<name>` and `<description>` are not defined in tab then `id` would be used for the display.

The Campaign/Offer/Creative/Channel/Slot/Placement/SlotType choice group performance tab has two sub tabs/pages; that is, Dashboard and Model Counts.

To include a Performance tab in a choice group, add the following code to the respective choice group's XML (such as `campaign.xml`, `channel.xml` and so on) in metadata folder; for example:

```
<tab id="Performance" type="report">
   <name>Performance</name>
   <description>Performance</description>
   <page id="PerformanceDashboard" viewType="view">
      <name>Dashboard</name>
      <description>Dashboard</description>
      <page-attributes-group>
         <bi-dashboard path=""/>
      </page-attributes-group>
   </page>
   <page id="ChoiceHistory" viewType="view">
      <name>Model Counts</name>
      <description>Model Counts</description>
      <page-attributes-group>
         <model-report reportType="choiceHistory"/>
      </page-attributes-group>
   </page>
</tab>
```

The following `<bi-dashboard>` metadata allows you to integrate the BI dashboard with in a tab. <path> is the path to the dashboard in the web catalog.

```
<bi-dashboard path="<path>">
   <parameter name="<name>" value="<value>"/>
   ...
</bi-dashboard>
```

`<path>` is the path to the dashboard in the web catalog; for instance, `path="/shared/Decision Analytics - Base Marketing/<ChoiceGroupName>/_portal/<ChoiceGroupName> Performance Dashboard">`.

The Campaign/Offer/Creative/Channel/Slot/Placement/Slottype/Tag choice group Analysis tab has four sub-tabs/pages; that is, Predictiveness, Best Fit, Drivers and

Model Quality. To include an Analysis tab in a choice group, add the following code to the respective choice group's XML (such as `campaign.xml`, `channel.xml` and so on) in metadata folder; for example.

```
<tab id="Analysis" type="report">
   <name>Analysis</name>
   <description>Analysis</description>
   <page id="PredReport" viewType="view">
      <name>Predictiveness</name>
      <description>Predictiveness</description>
      <page-attributes-group>
         <model-report reportType="predReport"/>
      </page-attributes-group>
   </page>
   <page id="BestFit" viewType="view">
      <name>Best Fit</name>
      <description>Best Fit</description>
      <page-attributes-group>
         <model-report reportType="bestFit"/>
      </page-attributes-group>
   </page>
   <page id="Drivers" viewType="view">
      <name>Drivers</name>
      <description>Drivers</description>
      <page-attributes-group>
         <model-report reportType="drivers"/>
      </page-attributes-group>
   </page>
   <page id="ModelQuality" viewType="view">
      <name>Model Quality</name>
      <description>Model Quality</description>
      <page-attributes-group>
         <model-report reportType="modelQuality"/>
      </page-attributes-group>
   </page>
</tab>
```

ref/project.xml

The Base Marketing (ref) `project.xml` file contains the extra attribute "type", and the pages associated with projects. In addition to the Overview page, which appears during project creation, editing, and viewing, the pages are defined for listing active and deployed projects.

ref/relationship-types.xml

The Base Marketing (ref) `relationship-types.xml` file describes the relationship types between the choice groups in the RTD for Marketing Optimization application.

For each relationship, the "from" and "to" choice groups are defined as well as the cardinality of the relationship.

In addition, there are properties to control the Delete logic (whether the delete applies just to one of the choice groups, or "cascades" to delete choices of the related choice group), rule propagation, and event propagation.

ref/security.xml

The Base Marketing (ref) `security.xml` file contains the configuration entries for the three application roles CLMConsumer, CLMAuthor, CLMAdministrator, and the choice group, project, and perspective permissions selected for those application roles. It contains no entries for any enterprise roles nor users.

ref/sql/insert ils data.sql and **ref/sql/insert marketing data.sql**

The insert ils data.sql and insert marketing data.sql files contain instructions to load some sample data for the Oracle RTD Decision Management database. These scripts are for demonstrations only.

10.3 Database Files for Core and Base Marketing

The database files for Core and Base Marketing in this section all appear under the directory **clm/Database**.

This section contains the following topics:

- sql/drop core.sql
- sql/load core.sql
- sql/ils/drop ils.sql
- sql/ils/load ils.sql

sql/drop core.sql

The sql/drop core.sql file contains instructions to drop all the generic metadata for a Oracle RTD Decision Management database - the tables, views, and sequences that provide the infrastructure for Oracle RTD Decision Management application data.

sql/load core.sql

The sql/load core.sql file contains instructions to create and initialize all the generic tables, views, and sequences required for a Oracle RTD Decision Management database. These tables, views, and sequences provide the infrastructure for Oracle RTD Decision Management application data.

sql/ils/drop ils.sql

The sql/ils/drop core.sql file, as released with Oracle RTD Base Application, contains instructions to drop all the tables and views specific to the individual choice groups defined in the released application RTD for Marketing Optimization.

Note: After choice groups or choice group attributes are added to, removed from, or have their definition properties altered in any Oracle RTD Decision Management application, and the application is then regenerated, the sql/ils/drop core.sql file is overwritten, and then reflects the new proposed data structure. The file is not automatically run after application regeneration.

---

sql/ils/load ils.sql

The sql/ils/load core.sql file, as released with Oracle RTD Base Application, contains instructions to create and initialize all the tables and views that implement the choice groups and the inter-choice group relationships defined in the released application RTD for Marketing Optimization.

Note: After choice groups or choice group attributes are added to, removed from, or have their definition properties altered in any Oracle RTD Decision Management application, and the application is then regenerated, the sql/ils/load core.sql file is overwritten, and then reflects the new proposed data structure. *The file is not automatically run after application regeneration.*