Contents

Preface ......................................................................................................................................................... xi
   Audience .............................................................................................................................................. xi
   Documentation Accessibility .............................................................................................................. xi
   Document Revision History .............................................................................................................. xi

Part I  Creating Pricing Components and Setup Components

1  About Creating Product Offerings
   About Pricing Components and Setup Components ........................................................................ 1-1
   Prerequisites for Creating Product Offerings .............................................................................. 1-3
   About Changesets ......................................................................................................................... 1-4
   Making Changes to Your Product Offerings .............................................................................. 1-4
      Deleting Charge Offers ............................................................................................................. 1-4
      Logging Changes to Product Offerings .................................................................................. 1-4
   About Product Offering Ownership .......................................................................................... 1-5

2  Creating Pricing Setup Components
   About Setup Components .............................................................................................................. 2-1
   Configuring Setup Components .................................................................................................. 2-2
   Configuring Ratable Usage Metrics (RUMs) ............................................................................. 2-4
   Setting up Service and Event Definitions .................................................................................. 2-5
      Using Service and Account Attributes with ECE for Usage Rating ...................................... 2-6
   Setting Up the Service-Event Map .............................................................................................. 2-7
   Configuring Balance Elements ................................................................................................... 2-8
      Enabling Single Currency Graph ............................................................................................. 2-9
   About Offer Profiles .................................................................................................................... 2-10

3  Synchronizing Pricing Setup Components
   About Synchronizing Setup Components ...................................................................................... 3-1
   About the SyncPDC Utility ............................................................................................................ 3-2
      Prerequisites for Running SyncPDC .......................................................................................... 3-2
   About Configuring SyncPDC ......................................................................................................... 3-2
      Updating the SyncPDCConfiguration.xml File ....................................................................... 3-7
   Running SyncPDC ....................................................................................................................... 3-7
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling Synchronization</td>
<td>3-7</td>
</tr>
<tr>
<td>Running SyncPDC Immediately</td>
<td>3-8</td>
</tr>
<tr>
<td>Running SyncPDC at a Scheduled Time</td>
<td>3-8</td>
</tr>
<tr>
<td>Running SyncPDC Immediately and at a Scheduled Time</td>
<td>3-8</td>
</tr>
<tr>
<td>Running SyncPDC at Recurring Scheduled Time</td>
<td>3-9</td>
</tr>
<tr>
<td>Synchronizing BRM Data with PDC to Use ECE for Usage Rating</td>
<td>3-9</td>
</tr>
<tr>
<td>Stopping SyncPDC</td>
<td>3-10</td>
</tr>
<tr>
<td>Generating Synchronization Reports</td>
<td>3-10</td>
</tr>
<tr>
<td>Handling Synchronization Failures</td>
<td>3-11</td>
</tr>
<tr>
<td>Reprocessing a Failed Synchronization</td>
<td>3-11</td>
</tr>
<tr>
<td>Changing Display Names</td>
<td>3-12</td>
</tr>
</tbody>
</table>

### 4 Configuring Charge Offers

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Charge Offers and Charges</td>
<td>4-1</td>
</tr>
<tr>
<td>Configuring Charge Offer Usage and Ownership</td>
<td>4-3</td>
</tr>
<tr>
<td>Restricting the End Time of Granted Balances That Start on First Usage</td>
<td>4-5</td>
</tr>
<tr>
<td>Charge Offers and Tax Calculation</td>
<td>4-6</td>
</tr>
<tr>
<td>Charge Offers and General Ledger</td>
<td>4-6</td>
</tr>
</tbody>
</table>

### 5 Configuring Charges in Charge Offers

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Charges</td>
<td>5-1</td>
</tr>
<tr>
<td>Specifying When a Charge is Effective</td>
<td>5-2</td>
</tr>
<tr>
<td>Applying a Charge to Inactive or Canceled Charge Offers</td>
<td>5-3</td>
</tr>
<tr>
<td>Configuring In-Advance Billing in Charges</td>
<td>5-3</td>
</tr>
<tr>
<td>Configuring Cycle Alignment for Recurring Charges</td>
<td>5-3</td>
</tr>
<tr>
<td>About Prorating Recurring Charges and Rollovers</td>
<td>5-3</td>
</tr>
</tbody>
</table>

### 6 Configuring Pricing in Charge Offers

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring Pricing in Charges</td>
<td>6-1</td>
</tr>
<tr>
<td>Configuring Validity Periods for Noncurrency Credit Balances</td>
<td>6-2</td>
</tr>
<tr>
<td>Specifying Rounding in Balance Impacts</td>
<td>6-3</td>
</tr>
<tr>
<td>Specifying if a Balance Impact is Discountable</td>
<td>6-3</td>
</tr>
<tr>
<td>About Minimum Charges</td>
<td>6-4</td>
</tr>
<tr>
<td>About Conditional Balance Impacts</td>
<td>6-4</td>
</tr>
<tr>
<td>About Prorating Charges</td>
<td>6-6</td>
</tr>
<tr>
<td>Configuring Dynamic Pricing</td>
<td>6-6</td>
</tr>
</tbody>
</table>

### 7 Pricing Configurations in Charge Offers

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring Pricing Based on Quantity</td>
<td>7-1</td>
</tr>
<tr>
<td>Configuring Effective Dates for Pricing</td>
<td>7-2</td>
</tr>
<tr>
<td>About Defining Pricing Based on Multiple RUMs</td>
<td>7-2</td>
</tr>
<tr>
<td>About Price Overrides</td>
<td>7-3</td>
</tr>
<tr>
<td>About Fold Charge Pricing</td>
<td>7-3</td>
</tr>
<tr>
<td>Granting First Usage Balances During Rating</td>
<td>7-4</td>
</tr>
</tbody>
</table>
8 Creating Discount Offers

About Discount Offers and Discounts
Configuring How Discount Offers are Purchased and Applied
   Applying Multiple Discount Offers to an Event
   Configuring Mutually Exclusive Discount Offers
Applying Discounts Based on Quantity
About Discount Filters
About Discount Triggers
About Using Temporary Balances in Discounts
Using Functions to Calculate a Balance Impact
Applying Multiple Discount Versions to an Event
About Billing-Time Discounts
Configuring Effective Dates for Discounts
Configuring Discount Validity
   Applying Discounts That Are Valid for Only Part of a Cycle
   Applying a Discount After the Discount Offer Is Inactivated or Canceled
   Configuring Validity Periods for Noncurrency Credit Balance Impacts
Discount Offers and Tax Calculation
Discount Offers and General Ledger
About Snowball Discounts
   Defining How Snowball Discounts Are Distributed

9 Creating Bundles

About Creating Bundles
About the Validity Periods of Offers in Bundles
About Bundles and First-Usage Balance Impacts
   About Activating First-Usage Discount offers
   About Granting First-Usage Balance Impacts In First-Usage Charges
   Aligning Validity of Balance Impacts on First Usage
Specifying Discounts in Bundles
Creating Dependencies for Bundles
Transitioning between Bundles

10 Creating Packages and Package Lists

Creating Packages
Grouping Services
Transitioning between Packages
   Defining Generation Changes
Creating Package Lists
   Specifying the Package List Segment and Type

11 Sharing Charges and Discounts

About Sharing Discounts and Charges
About First-Usage Start Time for Shared Balances
About Discount Sharing
Part II  Charging Scenarios

12  Charging Based on Date and Time

   About Charging Based on Date and Time ......................................................... 12-1
   About Special Day Calendars ........................................................................ 12-2
   Rating Events Split across Time Periods ....................................................... 12-3

13  Charging Based on Event Attributes

   About Impact Categories .................................................................................. 13-1
   About Zone Models .......................................................................................... 13-2
      Alternate Zone Models .................................................................................. 13-4
      Displaying Zoning Information on an Invoice ........................................... 13-4
   About Value Maps ............................................................................................ 13-4
   About Selectors .................................................................................................. 13-5
   About Usage Scenario (USC) Selectors ........................................................... 13-6
   About Mapping Services .................................................................................. 13-7
      Mapping Service Codes and Service Classes ............................................ 13-7
      Mapping Usage Classes ............................................................................. 13-7

14  Implementing Recurring Charges

   About Recurring Charges .................................................................................. 14-1
   About Cycle Forward Charges ......................................................................... 14-2
      About the First Cycle Forward Charge in a New Account ......................... 14-2
   About Cycle Arrears Charges ......................................................................... 14-2
   About Cycle Forward Arrears Charges ............................................................ 14-3
      About Implementing Cycle Forward Arrears Charging .............................. 14-4
   About Creating Recurring Charges in PDC .................................................... 14-4
      Specifying When Recurring Charges are Charged .................................... 14-4
      Allowing Recurring Charges to Be Prorated .............................................. 14-5
   About Recurring Charges Events ................................................................... 14-5

15  Advanced Configuration for Recurring Charges

   Configuring Flexible Cycle Forward Charges ............................................... 15-1
      Charging Cycle Forward Fees Associated with Flexible Cycles ................ 15-2
   Applying Cycle Forward Fees in Parallel ...................................................... 15-3
      About Enforcing Cycle Forward Fee Processing Prior to Billing ................ 15-4
      About Aggregating Service Charges to Account Items ............................. 15-4
      Handling Skipped Billing ............................................................................ 15-6
      Using the pin_bill_day Script to Apply Parallel Cycle Forward Fees ........ 15-6
      About Limitations and Impacts of Applying Cycle Forward Fees in Parallel  15-7
   Applying Recurring Charges Based on Charge Offer Priority ...................... 15-8
   Customizing the Cycle Interval for Specified Charge Offers ......................... 15-8
16  Working with Extended Rating Attributes
   About Extended Rating Attributes ................................................................. 16-1
   About Sharing ERAs ...................................................................................... 16-1
   Creating ERAs .............................................................................................. 16-1
   Creating Friends and Family ERAs ............................................................. 16-2
   Default ERAs ............................................................................................... 16-2

17  Configuring Closed User Groups
   Configuring Closed User Groups ................................................................ 17-1

18  Configuring Policy Specifications
   About Configuring Policy Specifications ...................................................... 18-1
   Configuring Policy Specifications ................................................................ 18-1

Part III  Managing Balances

19  Managing Balances during Charging
   Tracking Balances in Balance Groups .......................................................... 19-1

20  Managing Sub-Balances
   About Sub-Balances ................................................................................... 20-1
   About Noncurrency Sub-Balances .............................................................. 20-2
   How Balances in Validity-Based Sub-Balances Are Updated ......................... 20-3
   Restricting the Validity of Noncurrency Sub-Balances That Start on First Usage 20-4
   Configuring Custom Sub-Balances ............................................................... 20-5
   Defining and Loading Custom Sub-Balances ................................................. 20-7
   Specifying the Order in Which Sub-Balances Are Consumed ....................... 20-8
   Setting the Default Consumption Rule .......................................................... 20-10
   Configuring the Default Consumption Rule in ECE ..................................... 20-11
   Deleting Expired Sub-Balances .................................................................... 20-12
   Configuring Time-Stamp Rounding for Cycle Events .................................... 20-12
   Configuring Time-Stamp Rounding for Purchase Events .............................. 20-12

21  Configuring Item Type Selectors
   Configuring Item Type Selectors ................................................................. 21-1
   Loading the OOB_ItemTypeSelector.XML File .............................................. 21-5
   Configuring Item Type Selectors .................................................................. 21-5

22  Configuring Dynamic Quota
   Configuring Dynamic Quota ........................................................................ 22-1
   Configuring Dynamic Quota Selectors ......................................................... 22-5
## 23 About Folds

- About Folds .............................................................................................................. 23-1
- About Applying Folds Only For Account-Level Products .................................... 23-2

## 24 Applying Credit Limits to Balances

- About Credit Limits ........................................................................................................ 24-1
- About Credit Thresholds and Floors ......................................................................... 24-1
- Handling Credit Limit Conflicts .................................................................................. 24-2

## 25 About Rollovers

- About Rollovers ............................................................................................................. 25-1
- When Rollover Events Occur .......................................................................................... 25-2
- About Rolling Over Noncurrency Balances during Package Transition .................. 25-3
- About Rolling Over Balances That Expire in Midcycle ............................................. 25-3
- Prorating Rollover Balances ......................................................................................... 25-4
- Rollover Example .......................................................................................................... 25-5

## 26 Configuring Balance Impact Rounding

- About Balance Impact Rounding .................................................................................. 26-1
  - About Rounding Rules ............................................................................................... 26-1
  - How BRM Applies Rounding ...................................................................................... 26-2
    - About Rounding and A/R Actions ........................................................................... 26-4
    - About Rounding Billed and Unbilled Items ............................................................... 26-4
    - About Rounding for Specific Event Types ............................................................... 26-4
    - About Rounding Counter Balances for Discounting .............................................. 26-4
    - About G/L Report Rounding ..................................................................................... 26-5
  - About Rounding Modes ............................................................................................... 26-5
    - About Rounding Modes That Correct for Loss of Precision .................................... 26-6
  - When Rounding Is Not Applied ................................................................................. 26-7
- Configuring Balance Impact Rounding ......................................................................... 26-7
  - About Configuring Rounding Rules .......................................................................... 26-7
  - About Rounding Mode Values in the BRM API ......................................................... 26-7
  - Configuring Rounding Rules for Counter Balances ................................................ 26-8
    - About Rounding Modes for Discount Expressions .................................................. 26-8
  - Configuring to Record Rounding Differences in the G/L ......................................... 26-8
    - Defining a G/L ID for Rounding Differences ............................................................. 26-9
    - Mapping the Rounding G/L ID to an Event .............................................................. 26-9
    - Configuring BRM to Record Rounding Differences ................................................ 26-10
  - Rounding Examples .................................................................................................... 26-11
    - Correcting for Precision Loss When Rounding Down ............................................. 26-11
    - Rounding Using Different Modes ............................................................................ 26-11
- Configuring Balance Impact Rounding in ECE ........................................................... 26-13
  - Configuring Rounding for a Currency Balance Impact ............................................ 26-14
  - Configuring Rounding for a Noncurrency Balance Impact ...................................... 26-15
  - Configuring Rounding When Authorizing Multiple RUMs ...................................... 26-16
Part IV  Customization and Administration

27  Enabling Charging for Custom Events
Overview of Enriching Event Definitions ................................................................. 27-1
About Charging Operation Types ........................................................................... 27-2
About Charging Operation Type Versions ............................................................ 27-2
About Mapping Network Attributes to Event Attributes ....................................... 27-3
Before Enriching Event Definitions ..................................................................... 27-3
Enriching Event Definitions .................................................................................. 27-4
Editing Event Definitions ...................................................................................... 27-5

28  Using the PDC Web Service
About the PDC Web Service API .......................................................................... 28-1
About Using the PDC Web Service ........................................................................ 28-1
About Creating the XML Files ............................................................................ 28-2
About Using the PDC Web Service to Create or Modify Pricing Components ........ 28-2
CreatePricing ........................................................................................................... 28-4
ModifyPricing ......................................................................................................... 28-4
CreatePricingAndSubmit ....................................................................................... 28-4
ModifyPricingAndSubmit ...................................................................................... 28-5

29  Working with Profiles
About Pricing Profiles ............................................................................................ 29-1
About Default Pricing Profiles ............................................................................. 29-1
About Customizing Pricing Profiles ...................................................................... 29-2
About Pricing Types ............................................................................................... 29-3
About the PDC Target Engine Profiles ................................................................. 29-4
BRM Online Rating Target Engine Profile ............................................................ 29-5
BRM Offline Rating Target Engine Profile ............................................................ 29-6
Subscription Engine Profile .................................................................................. 29-8
ECE Rating Target Engine Profile ........................................................................ 29-8
ECE Subscription Engine Profile .......................................................................... 29-10
Target Engine Profiles for the Default Pricing Profiles ....................................... 29-10
About Associating Custom Pricing Profiles with Target Engine Profiles ............. 29-11
About Importing Custom Pricing Profiles into PDC ............................................ 29-12
Setting Up Custom Pricing Profiles ..................................................................... 29-12
Creating Custom Pricing Profiles ......................................................................... 29-12
Associating Custom Pricing Profiles with Target Engine Profiles ...................... 29-13
Importing Custom Pricing Profiles into PDC ....................................................... 29-13

30  Importing and Exporting Pricing and Setup Components
About Using the XML Pricing Interface to Create Product Offerings ................. 30-1
About the ImportExportPricing Utility ................................................................. 30-2
About Dependencies between Pricing and Setup Components .......................... 30-2
About the XSD Files .............................................................................................. 30-4
Prerequisites for Using the ImportExportPricing Utility .......................................................... 30-4
Exporting Pricing and Setup Components from PDC ............................................................ 30-4
   Exporting All the Pricing and Setup Components ................................................................. 30-4
   Exporting a Subset of Pricing and Setup Components ......................................................... 30-5
      Exporting Only the Setup Components that are Mastered in PDC .................................. 30-5
      Exporting Only the Setup Components that are Mastered in BRM ................................. 30-6
      Exporting All Pricing Reference Objects ..................................................................... 30-6
      Exporting Only the Metadata Objects .......................................................................... 30-6
      Exporting Setup Components Modified After a Specific Time ...................................... 30-6
      Exporting Pricing Components Modified by Specific Users ........................................ 30-6
      Exporting Only Successfully Promoted and Transformed Components ...................... 30-6
      Exporting Only Obsoleted Components ....................................................................... 30-7
   About Supported Pricing and Setup Components ............................................................... 30-7
Importing Pricing and Setup Components ........................................................................... 30-9
Listing Pricing or Setup Components .................................................................................. 30-9
Publishing Pricing and Setup Components ......................................................................... 30-10
Deleting Components Mastered in BRM .......................................................................... 30-10
   Deleting Pricing Profiles ................................................................................................. 30-10
   Deleting Old Versions of PDC Components .................................................................. 30-11
   Obsoleting PDC Components ......................................................................................... 30-11
   Ignoring Internal IDs ....................................................................................................... 30-12
Creating the Pricing and Setup Components by Using XML Files ...................................... 30-12
   Creating the XML Files ................................................................................................. 30-12

31 Replicating Data Between PDC Systems
   About Replicating Data ..................................................................................................... 31-1
   Replicating Setup Components Mastered in BRM .......................................................... 31-1
   Replicating Data Between PDC Databases ...................................................................... 31-2

32 Pricing Utilities
   ImportExportPricing .......................................................................................................... 32-2
   PDCWalletUtil.sh ............................................................................................................. 32-12
   PDCBRMWalletUtil.sh ..................................................................................................... 32-13
   load_pin_snowball_distribution ...................................................................................... 32-14
   load_pin_spec_rates ......................................................................................................... 32-16
   load_pin_sub_bal_contributor ......................................................................................... 32-17
   pin_discount_cleanup ...................................................................................................... 32-19
   New Pricing Terminology ............................................................................................... A-2
Preface

This guide describes how to create product offerings for Oracle Communications Billing and Revenue Management (BRM) in Pricing Design Center (PDC).

Audience

This guide is intended for pricing analysts and others involved in pricing, rating, and rerating.

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 that have purchased support 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.

Document Revision History

The following table lists the revision history for this book:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E51022-01</td>
<td>December 2017</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>
### Documentation updates for PDC 12.0 Patch Set 1

1. Added the following chapters:
   - Configuring Dynamic Quota
   - Configuring Policy Specifications

2. Updated the following sections:
   - Exporting Pricing and Setup Components from PDC
   - Deleting Old Versions of PDC Components
   - Obsoleting PDC Components
   - Exporting Pricing and Setup Components from PDC
   - ImportExportPricing

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>E51022-02</td>
<td>September 2018</td>
<td>Documentation updates for PDC 12.0 Patch Set 1.</td>
</tr>
</tbody>
</table>

- Added the following chapters:
  - Configuring Dynamic Quota
  - Configuring Policy Specifications
- Updated the following sections:
  - Exporting Pricing and Setup Components from PDC
  - Deleting Old Versions of PDC Components
  - Obsoleting PDC Components
  - Exporting Pricing and Setup Components from PDC
  - ImportExportPricing
Part I

Creating Pricing Components and Setup Components

This part describes how to create Oracle Communications Billing and Revenue Management (BRM) pricing components and setup components in Pricing Design Center (PDC).

Part I contains the following chapters:

- About Creating Product Offerings
- Creating Pricing Setup Components
- Synchronizing Pricing Setup Components
- Configuring Charge Offers
- Configuring Charges in Charge Offers
- Pricing Configurations in Charge Offers
- Creating Discount Offers
- Creating Bundles
- Creating Packages and Package Lists
- Sharing Charges and Discounts
About Creating Product Offerings

This document provides an overview of creating product offerings for Oracle Communications Billing and Revenue Management (BRM) in Pricing Design Center (PDC).

Topics in this document:

■ About Pricing Components and Setup Components
■ Prerequisites for Creating Product Offerings
■ About Changesets
■ Making Changes to Your Product Offerings
■ About Product Offering Ownership

See also:

■ BRM Concepts
■ Configuring Charge Offers
■ Creating Discount Offers

PDC includes sample product offerings for a broadband service and for a telco service in the PDC_home/apps/samples/data directory, where PDC_home is the directory in which PDC is installed.

About Pricing Components and Setup Components

The product offerings you create are composed of pricing components and setup components.

Pricing components enable you to sell your products and services to customers. They specify how much to charge, and how the services are made available to your customers. The two main pricing components are charge offers and discount offers.

■ Charge offers specify how to charge your customers for your services.
■ Discount offers specify when and how to reduce the cost of an event or to change the balance of an element such as minutes or loyalty points.

You package charge offers and discount offers in a hierarchy of bundles, packages, and package lists:

■ Bundles contain one or more charge offers, discount offers, or both.
■ Packages contain one or more bundles. You use packages to offer one or more services to customers. For example, you can offer one service with two different ways of charging for it, each defined in a separate bundle.
Package lists contain a group of packages that are offered to a single type of customer, for example, one package list for customers who create and account by using the Web, and another package list for customers who create and account by talking to a CSR.

Chargeshare offers and selectors are less commonly used product components:

- **Chargeshare offers** provide the ability for an account to assume all or part of the charges for services owned by other accounts. For example, a business may want to pay for some or all of the wireless charges for its employees. Chargeshare offers specify which events qualify for charge sharing and the conditions that must be met for charge sharing to apply.

- **Selectors**: contain rules that enable the same charge offer or discount offer to apply different balance impacts based on event attributes. For example, a charge selector can determine which charge to use based on the call origins and destinations recorded in events.

Setup components are the prerequisite data used to create pricing components. The basic setup components that are required for all product offerings are the following:

- **Ratable usage metrics (RUMs)** specify how to measure events (for example, by duration or occurrence). For example, you could charge for the occurrence of a video download as well as for the duration.

- The **service-event map** links each service your system supports to the events that can have charges configured for that service. For example, for a GSM mobile service, you can specify to rate usage events.

  Each charge offer applies to a particular service. When you create a charge offer, you select events related to that service to configure charges for. To prevent you from selecting an event that does not occur for a service, PDC uses the service-event map.

- **Balance elements** specify attributes for one of the following:
  - A currency or noncurrency asset, such as U.S. dollars or included minutes. A balance element contains rules for rounding and consuming the asset and an option to use it as a temporary balance element.
  - A counter that tracks items such as dollars spent or minutes talked.

You can define these setup components for additional charging functionality:

- **Time Models** contain a set of time periods to enable charging different prices for the same service depending on the day and time the service is used.

- **Special day calendars** define a set of dates, such as holidays, for which you want to charge special prices for your services. These dates are used in a time model to define a time period. A time period can be used in a charge offer to determine a price.

- **Impact categories** enable the same charge to apply different pricing based on the values of various event attributes. For example, to configure different pricing for calls made to different countries, you add impact categories for each destination country to the charge. When a call occurs, the pricing associated with the impact category for that call's destination is applied to the call. See "Charging Based on Event Attributes" for more information.

- **Zone models** contain a set of zones which enable charging for calls based on their origin and destination

  You can configure the following types of zone models:
Prerequisites for Creating Product Offerings

Before you create product offerings, you must complete the following tasks.

- **Customize services and events**
  You might need to add or modify services before you create your product offerings. If you add services, you need to create events to track the services.

- **Define custom account attribute data**
  To structure pricing, you might want to create charges based on account information; for example, the location of a customer and the type of account (such as Premium or Standard). If you use custom account attributes to base charging on, you need to create those custom attributes. For example, you might define charges based on a subscriber’s type of employment.

- **Define tax codes and tax suppliers**
  A tax code is a code that identifies a type of charge offer, such as a subscription or a physical commodity. You map tax codes to taxation packages that calculate the tax on the charge offers or to custom tax rates that you define. Tax codes are included in charge offers to indicate which tax to apply.

  A tax supplier is a company or company division responsible for collecting taxes. Taxes can be calculated differently depending on the location of the tax supplier.

- **Create G/L IDs**
  You use G/L IDs to collect general ledger information from your database and to export it to your accounting application. You must decide how to track the revenue for each type of charge and create the appropriate G/L IDs. You assign G/L IDs in charge offers.

  G/L ID is an ID assigned to the balance impact. The G/L ID ensures that the balance impact is reported to the correct account in your company’s G/L. For example, you keep a record of revenue from usage fees by mapping usage balance impacts to G/L IDs.

- **Define provisioning tags**
  If you use charge offer or discount offer provisioning, you must define provisioning tags. Provisioning tags define the custom provisioning tasks that...
need to be performed to implement a service on the network; for example, create a voice mailbox.

**About Changesets**

In PDC, you create and modify all setup and pricing components in the context of a changeset. A changeset tracks the changes that you make to setup components and pricing components. Each user has one or more changesets, which only that user can access.

If you do not have a changeset, PDC automatically creates one for you when you log in. This changeset becomes your active changeset. When you create a setup component or a pricing component, a draft of the new component is added to your active changeset. When you modify a component, the updated component remains in the changeset in which it was created. If you have multiple changesets, only one can be active, but you can designate a different changeset as your active changeset at any time.

To publish the pricing components in a changeset to the BRM database, you submit the changeset. You must submit an entire changeset; you cannot submit only some of its components.

You can import components from an XML file into a new or an existing open changeset. The imported components can be immediately submitted for publication.

To share your changesets with other users, you can export the components from a changeset into an XML file. Optionally, you can include all the components referenced by the components in the exported changeset, whether or not the components are part of that changeset.

**Making Changes to Your Product Offerings**

You can make changes to your product offerings at any time. For example, if you offer a new service, you can create new product offerings to charge for that service.

You can also change charges for existing charge offers, add charge offers to existing bundles, and so forth.

**Deleting Charge Offers**

You cannot delete a charge offer if it is owned by any account. To delete a charge offer, cancel it in all accounts.

**Logging Changes to Product Offerings**

To get notified when product offerings change, BRM can create messages in the `cm.pinlog` file when product offerings are updated. For information about `cm.pinlog` and other log files, see "Using logs to monitor components" in *BRM System Administrator’s Guide*.

To log product offering changes, set the `fm_rate log_refresh_product` entry in the Connection Manager (CM) `pin.conf` to 1. If this entry is absent or set to 0, BRM does not log changes to the product offerings.
To log product offering changes:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Set the value for the `fm_rate log_refresh_product` entry to 1.
   - `fm_rate log_refresh_product 1`
3. Save the file.
4. Stop and restart the CM.

**About Product Offering Ownership**

During account creation, the customer chooses a package. A package is a set of bundles, each of which in turn is a set of charge offers. Therefore, what the customer actually owns is not a package, but a set of charge offers and discount offers.

If you create different charge offers for the same service, different customers might use the same service, but pay different charges, based on the charge offers that they own.

For example, you might have two customer accounts that use the same service, but have different charge offers as shown in Table 1–1:

<table>
<thead>
<tr>
<th>Account 1</th>
<th>Account 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadband charge offer 1:</td>
<td>Broadband charge offer 2:</td>
</tr>
<tr>
<td>■ Service: IP</td>
<td>■ Service: IP</td>
</tr>
<tr>
<td>■ Cycle forward event: $20</td>
<td>■ Cycle forward event: $10</td>
</tr>
<tr>
<td>■ Usage events: $2</td>
<td>■ Usage events: $1</td>
</tr>
</tbody>
</table>

When BRM rates events for these customers, the charges for Account 1 are different from the charges for Account 2, even though the services and the events being rated are the same type.

When you manage a customer’s account, you manage their charge offers. For example, you can change the charge offer status or customize the charge offer’s pricing. When you create your product offerings, it is important to consider how the charge offers will be managed after they have been purchased.
Creating Pricing Setup Components

This document describes how to create the Oracle Communications Billing and Revenue Management (BRM) pricing setup components in Pricing Design Center (PDC).

Topics in this document:

- About Setup Components
- Configuring Setup Components
- Configuring Ratable Usage Metrics (RUMs)
- Setting up Service and Event Definitions
- Setting Up the Service-Event Map
- Configuring Balance Elements
- About Offer Profiles

See also:

- About Creating Product Offerings
- Configuring Charge Offers

About Setup Components

Setup components are the prerequisite data that you configure before creating product offerings. For example, before creating product offerings, you must configure the ratable usage metrics (RUMs) that define how to measure events.

Some setup components are defined in PDC and some components are mastered in Oracle Communications Billing and Revenue Management (BRM) and need to be loaded into PDC.

The following setup components are defined in PDC. Any modifications to these components are also handled by PDC and are transformed to the BRM database and ECE:

- Ratable Usage Metrics (RUMs)
- Service-event maps
- Balance elements
- Impact categories
- Zone models
- Value maps
Configuring Setup Components

You configure the setup components in PDC by using the following applications:

- **PDC UI.** Use to create and modify setup components using a graphical user interface. See the PDC Help for more information.

- **ImportExportPricing utility.** Use to import setup components defined in an XML file into PDC. See "Importing and Exporting Pricing and Setup Components" for more information.

- **SyncPDC utility.** Use to synchronize setup components defined in BRM with PDC. See "Synchronizing Pricing Setup Components" for more information.

Table 2–1 lists the setup components, the order in which you should configure the data, whether the setup component is required or optional, and the application that you use to configure them.

<table>
<thead>
<tr>
<th>Pricing Setup Data</th>
<th>Required/Optional</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUMs</td>
<td>Required</td>
<td>PDC UI or ImportExportPricing utility</td>
</tr>
<tr>
<td>Service-event maps</td>
<td>Required</td>
<td>PDC UI or ImportExportPricing utility</td>
</tr>
<tr>
<td>Impact categories</td>
<td>Optional</td>
<td>PDC UI or ImportExportPricing utility</td>
</tr>
<tr>
<td>Zone models</td>
<td>Optional</td>
<td>PDC UI or ImportExportPricing utility</td>
</tr>
</tbody>
</table>

Note: Some of the pricing setup data is optional. You must configure the optional pricing setup data only if you plan to use the related feature or perform certain business functions. For example, you configure a zone model only if you use zoning to determine a price in your charge offer.
Sample XML files for each setup component are in the `PDC_home/apps/Samples/Examples` directory, where `PDC_home` is the directory in which the PDC software is installed.

The `Examples` directory contains the following sample and ready-to-use files:

- **PDCSampleBE.xml.** Contains all currency balance elements and some frequently used noncurrency balance elements. You can load this file to quickly create balance elements to use when configuring pricing.

- **PDCSampleRum.xml.** Contains commonly used ratable usage metrics (RUMs). You can load this file to quickly create RUMs to use when creating a service-event map and when configuring pricing.

- **OOB_ProfileSpecifications.xml.** Contains profile attribute specifications referenced in the following custom rules: `Friends&Family`, `SpecialDay`, and `ClosedUserGroup`. This XML file references the names used in the provisioning tags provided by default with the BRM installation.

  **Note:** You must load the `OOB_ProfileSpecifications.xml` file before loading the `OOB_CRs.xml` file.

- **OOB_CRs.xml.** Contains the custom rules: `Friends&Family`, `SpecialDay`, and `ClosedUserGroup`. You can use these custom rules in a generic selector when

### Table 2–1 (Cont.) Setup Components and the Application You Use to Configure Them

<table>
<thead>
<tr>
<th>Pricing Setup Data</th>
<th>Required/Optional</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Special day calendars</td>
<td>Optional</td>
<td>PDC UI or ImportExportPricing utility</td>
</tr>
<tr>
<td>Services</td>
<td>Required</td>
<td>SyncPDC utility or ImportExportPricing utility</td>
</tr>
<tr>
<td>Events</td>
<td>Required</td>
<td>SyncPDC utility or ImportExportPricing utility</td>
</tr>
<tr>
<td>Account attributes</td>
<td>Required</td>
<td>SyncPDC utility or ImportExportPricing utility</td>
</tr>
<tr>
<td>Tax codes</td>
<td>Optional</td>
<td>SyncPDC utility or ImportExportPricing utility</td>
</tr>
<tr>
<td>Tax suppliers</td>
<td>Optional</td>
<td>SyncPDC utility or ImportExportPricing utility</td>
</tr>
<tr>
<td>G/L IDs</td>
<td>Required for some target engines</td>
<td>SyncPDC utility or ImportExportPricing utility</td>
</tr>
<tr>
<td>Provisioning tag names</td>
<td>Optional</td>
<td>SyncPDC utility or ImportExportPricing utility</td>
</tr>
<tr>
<td>Balance elements</td>
<td>Required</td>
<td>ImportExportPricing utility</td>
</tr>
<tr>
<td>Value maps</td>
<td>Optional</td>
<td>ImportExportPricing utility</td>
</tr>
<tr>
<td>Custom rules</td>
<td>Required for some target engines</td>
<td>ImportExportPricing utility</td>
</tr>
<tr>
<td>Profile attribute specifications</td>
<td>Required for some target engines</td>
<td>ImportExportPricing utility</td>
</tr>
<tr>
<td>Item type selector</td>
<td>Required for ECE</td>
<td>ImportExportPricing utility</td>
</tr>
<tr>
<td>Business Profile</td>
<td>Required for ECE</td>
<td>SyncPDC utility</td>
</tr>
<tr>
<td>Policy Specifications</td>
<td>Optional</td>
<td>PDC UI or ImportExportPricing utility</td>
</tr>
</tbody>
</table>
configuring a charge based on a subscriber’s Friends & Family, Special Day, or Closed User Group profile.

- **Sample_ServiceCUG_ProfileSpecification.xml.** Contains the profile attribute specification for configuring a service-level closed user group. See “Configuring Closed User Groups” for more information.

- **Sample_ServiceCUG_CR.xml.** Contains the custom rule for configuring a service-level closed user group. See “Configuring Closed User Groups” for more information.

- **OOB_ItemTypeSelector.xml.** Contains the default billing item assignments that are supported by BRM without any additional configuration. You must load this file if you are using ECE. See "Loading the OOB_ItemTypeSelector.XML File" for more information.

- **Sample_ItemTypeSelector.xml.** Contains the item type selector rules for defining separate bill items for international and national calls. You can use this as a template to quickly create the item type selector for tracking international calls and national calls separately. See "Configuring Item Type Selectors" for more information.

- **SamplePolicyLabel.xml.** Contains policy labels referenced in the policy specifications. You can use this as a template to add and modify the elements as required. See “Configuring Policy Specifications”.

- **SamplePolicySpecification.xml.** Contains policy specifications referenced in the charge offers and discounts offers used for policy-driven charging. You can use this as a template to add and modify the elements as required. See "Configuring Policy Specifications”.

You can load the updated sample XML files into PDC by using the **ImportExportPricing** utility. See "ImportExportPricing" for the utility’s syntax and parameter descriptions.

---

**Note:** The Examples directory also contains sample pricing profile files that are provided only for reference and for use as templates for creating custom pricing profiles. You should not load the sample pricing profile files after installing PDC.

---

## Configuring Ratable Usage Metrics (RUMs)

RUMs are ways in which you can measure events. The most common RUMs are Duration and Volume.

Before you create product offerings, you must define the RUMs available for rating. You define RUMs when you create the service-event map. For each event, you specify the RUMs that can be used for rating it.

When you define RUMs, you define the following RUM attributes:

- **The unit of measurement.** For example, to rate duration, you might specify seconds or minutes. To measure bytes, you might specify megabytes or kilobytes.

- **How to calculate the quantity.** In PDC, use the RUM expression. For example, if you specify Minute as the unit of measurement, and you specify 1 as the expression, the duration is calculated in 1-minute increments.
You can also perform calculations on fields in the event by using the following operators:

- Add: +
- Subtract: -
- Multiply: *
- Divide: /

### How to round quantities:

- Half Up: A fraction of .5 or greater is rounded up. For example, 1.5 is rounded to 2.
- Half Down: A fraction of .5 or less is rounded down. For example, 1.5 is rounded to 1.
- None
- Round Up: Any fraction is always rounded up
- Round Down: Any fraction is always rounded down

### Type:

- Whole: Select this type if the RUM cannot be split; for example, Occurrence.
- Partial: Select this type if the RUM can be split; for example, Duration.
- Conditional: Select this type to configure charges with conditional balance impacts.

---

## Setting up Service and Event Definitions

BRM provides a preconfigured set of service and event definitions. When you run the SyncPDC utility, the preconfigured service and event definitions are stored in PDC. For the list of preconfigured events, see the `BRM_Home/sys/data/pricing/example/pin_event_map` file, where `BRM_Home` is the directory in which you installed BRM.

**Note:** Before you run the SyncPDC utility for the first time, ensure that only those services and events that are relevant to your business are available in the `pin_event_map` file.

---

If you are using ECE for usage rating, the preconfigured event definitions are automatically enriched with required information for ECE when you run the SyncPDC utility. You can use these definitions for pricing and charging without any further enrichment.

Enriching includes adding user-defined charging attributes, defining charge operation types, and mapping event attributes to the external network format. For example, you add specific attributes that ECE uses to apply session-based charge for a transaction.

You can also create your own service and event definitions:

- To offer new services and events
- To add additional attributes for provisioning, networking, charging, or billing

You can define the custom services and events in Development Center and use the SyncPDC utility to synchronize the definitions with PDC. If you are defining a new service or event, ensure that you do the following:
1. Create a subclass for the new service or event in Development Center. See the discussion about creating storable classes for services and events in BRM Developer’s Guide for more information.

2. Create a substruct in the event subclass; for example, USAGE_INFO, and add USERIDENTITY and CALLED_ID fields in Development Center. See the discussion about substructure and creating custom fields in BRM Developer’s Guide for more information.

3. Map the event to the corresponding service by editing the pin_event_map file and then running the load_event_map utility. See the discussion about mapping event types to services in BRM Setting Up Pricing and Rating for more information.

4. Synchronize the service and event definitions with PDC by running the SyncPDC utility. See “Synchronizing Pricing Setup Components” for more information.

   The service and event definitions are loaded into PDC.

5. If you are using ECE for usage rating, enrich the event definitions with the information ECE requires for processing usage requests. See “Enabling Charging for Custom Events” for more information.

   The enriched event definition is published to ECE.

6. If you are using ECE for usage rating and you want to use any service or account attributes for usage rating, ensure that only those service or account attributes are loaded into the ECE cache. See “Using Service and Account Attributes with ECE for Usage Rating” for more information.

### Using Service and Account Attributes with ECE for Usage Rating

If you are using ECE for usage rating and you want to use any service or account attributes for usage rating, ensure that only those service or account attributes are loaded into the ECE cache by doing the following:

1. Export the service or account definition into an XML file by running the following command:

   ```
   ImportExportPricing -export -metadata -n PRODUCT_ATTRIBUTE_SPEC|CUSTOMER_ATTRIBUTE_SPEC "ObjectName"
   ```

   where `ObjectName` is the name of the service or account to be exported into an XML file.

   For example:

   ```
   ImportExportPricing -export -metadata PRODUCT_ATTRIBUTE_SPEC -n TelcoGSM
   ```

   Exports the definition for TelcoGSM to the `productattr_export_config.xml` file.

2. Open the XML file in a text editor.

3. Search for the attributes that you want to use for usage rating and add the following entries for each attribute:

   ```
   <attributeItemSupportedBy>
       <targetApplicationSpecName>Pricing</targetApplicationSpecName>
       <targetApplicationSpecName>Convergent Charging</targetApplicationSpecName>
   </attributeItemSupportedBy>
   ```

4. Save and close the file.

5. Import the XML file into PDC by running the following command:
ImportExportPricing -import -config FileNamePrefix_export_config.xml ow

For example:

ImportExportPricing -import -config productattr_export_config.xml

The service or account definition is loaded into PDC. The ECE Pricing Updater publishes only the service or account attributes that you want to use for usage rating to ECE and only those attributes are loaded into the ECE cache.

Setting Up the Service-Event Map

Before you can create charge offers, you must set up a service-event map. The service-event map lists all the services to which charge offers can apply. For each service, the map specifies which events can have charges configured for that service. The map also specifies the RUMs to use for rating each event combination.

For events that do not apply to a specific service, you map those events to Account in the service-event map.

For example, in Figure 2–1, the events Cancel Fee, Cycle Fold, GSM Session, Monthly Recurring, and Monthly Recurring Arrear are mapped to the GSM service. The RUMs, Duration and Volume, are mapped to the GSM Session event.
Configuring Balance Elements

Before you can create a pricing component, you must use PDC to create balance elements. You can create currency and noncurrency balance elements.

When you create balance elements, you define the following:

- The balance element name, such as US Dollars.
- The balance element ID, such as 840. Currency balance element IDs are defined in an ISO standard.
- The rounding value. For example, to round US dollars to cents, specify a rounding value of two places after the decimal point.
- How to round additional numbers beyond the rounding value:
  - **Up**: For example, 10.151 rounds to 10.16.
  - **Down**: For example, 10.159 rounds to 10.15.
Nearest: If the additional digit is 0-4, the last significant digit remains the same. If the additional digit is 5-9, the last significant digit is rounded up. For example, 10.144 rounds to 10.14 and 10.145 rounds to 10.15.

Even: If the additional digit is 0-4, the last significant digit remains the same. If the additional digit is 6-9, the last significant digit is rounded up. If the additional digit is 5, the last significant digit is rounded to the nearest even digit. For example, if rounding is set to two significant digits, 10.155 rounds to 10.16 and 10.165 rounds to 10.16.

See "Configuring Balance Impact Rounding" for more information about rounding balance elements.

For currency balance elements, you also define the error tolerance and the abbreviations and symbols used for display, such as $.

The order in which to consume sub-balances. For example, if a customer’s balance contains minutes with different validity periods, you specify which minutes to use first, according to the starting validity date or ending validity date. See "Specifying the Order in Which Sub-Balances Are Consumed".

You can add multiple currency balance elements for subscriber currencies under a single pricing tier by using the ImportExportPricing utility. See "Enabling Single Currency Graph" for more information.

Enabling Single Currency Graph

To enable a single currency graph:

1. Go to the PDC_Home/apps/bin directory.
2. Export the pricing data by running the ImportExportPricing utility.

For example, to export the data to a file named chargeoffer_pricing.xml, run the following command:

```
./ImportExportPricing -export chargeoffer -pricing
```

3. In the chargeoffer_pricing.xml file, search for the balanceElementNumCode attribute. By default, only one currency code is available. You can add more currency codes.

For example:

```xml
  -<subscriberCurrency>
    <currencyCode>USD</currencyCode>
    <currencyCode>EUR</currencyCode>
  -<applicableRum>
    <applicableRumName>Duration</applicableRumName>
    <minQuantity>1.0</minQuantity>
    <minQuantityUnit>NONE</minQuantityUnit>
    <incrementQuantity>1.0</incrementQuantity>
    <incrementQuantityUnit>NONE</incrementQuantityUnit>
    <roundingMode>NEAREST</roundingMode>
    -<crpRelDateRange>
      -<absoluteDateRange>
        <startDate>0</startDate>
        <endDate>inf</endDate>
```

Note: Single currency graph can be used only with ECE.
4. Save the XML file as `import_pricing.xml` file.

5. Import the `import_pricing.xml` file by using the `ImportExportPricing` utility. For example:

   ```bash
   .;/ImportExportPricing -import -pricing import_pricing.xml -ow
   ```

6. Add, edit, or remove balance impacts for all the supported subscriber currencies.

**About Offer Profiles**

Offer profiles are made up of one or more policy labels each of which defines a gradation in the quality of service (QoS) based on usage amounts.

For example, you can have an offer profile called "Platinum" for a data service and define its balance element as *Megabytes Used*. You can define a policy labeled *Fair Usage*, which has three levels, *Low QoS*, *Medium QoS*, and *High QoS*, with each level containing a usage range valid for that quality of service.
This chapter describes how to synchronize Oracle Communications Billing and Revenue Management (BRM) setup components with Pricing Design Center (PDC).

About Synchronizing Setup Components

When configuring a new PDC system, you use the SyncPDC utility to synchronize the setup components defined in billing and rating systems with PDC. See "Configuring Setup Components" for more information.

**Note:** Ensure that `/event/runtimeDiscount` does not exist in BRM during synchronization.

SyncPDC synchronizes the following setup components:

- Service definitions
- Event definitions
- Account definition
- General ledger (G/L) IDs
- Provisioning tags
- Tax codes
- Tax suppliers
- Business profiles

After these setup components are synchronized with PDC, any modifications to these setup components must only be done in the system in which they were initially defined and then resynchronized.

SyncPDC synchronizes only the changes from the previous synchronization. It generates reports for the components that it synchronizes. See "Generating Synchronization Reports" for more information.

When synchronizing setup components, SyncPDC creates display names in PDC for the corresponding data objects and their fields. For example, it creates the **GsmTelephony** PDC name for the `/service/gsm/telephony` service name in BRM. You can change the PDC display name in PDC. For example, you can change **EventTelcoGsmVoice** to **Voice Call**. If you later run SyncPDC to resynchronize this event, it retains the name change. See "Changing Display Names" for more information about updating PDC display names.
If you delete a setup component (or any of its fields) from the billing or rating system, you must manually remove the corresponding PDC component (or its fields) from the PDC database. See "Deleting Components Mastered in BRM" for more information.

About the SyncPDC Utility

The SyncPDC utility runs as a server process in the background, continuously checking for data to synchronize from BRM or rating system with PDC. You can schedule it to run immediately, at a specified time, or regularly at a specified time. At any given time, you can stop the synchronization by stopping SyncPDC. See "Stopping SyncPDC" for more information.

Prerequisites for Running SyncPDC

Before you run SyncPDC, do the following:

- Start the transformation engines for the rating systems. See the discussion of starting the transformation engines in PDC Installation Guide.
- Include the \PDC_home\apps\bin directory in your PATH environment variable, where \PDC_home\ is the directory in which you installed the PDC software.
- Configure the \BRM_Integration_Pack_\Home\apps\syncpdc\SyncPDCConfiguration.xml file. See Table 3–1, "Elements in the SyncPDCConfiguration.xml File", for more information.

To run SyncPDC, you must have Pricing Design Admin role privileges.

About Configuring SyncPDC

SyncPDC uses the SyncPDCConfiguration.xml file, which contains connection information for the Oracle WebLogic Server Administration, the PDC server, and the cross-reference database and configuration settings for SyncPDC. This file is generated during PDC installation and is located in the \BRM_Integration_Pack_\Home\apps\syncpdc directory. By default, this file contains the values that you provide during BRM Integration Pack installation. You can change the default values by updating this file.

Table 3–1 lists the elements in SyncPDCConfiguration.xml, the usage of each element, and a description of how to specify each element based on the default version of the file.
### Table 3–1 Elements in the SyncPDCConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration</td>
<td><code>&lt;Configuration&gt;</code></td>
<td>The root element of SyncPDCConfiguration.xml.</td>
</tr>
<tr>
<td>walletConfiguration</td>
<td><code>&lt;walletConfiguration&gt;</code></td>
<td>Specifies the Oracle wallet that stores the sensitive information, such as database passwords, where BRMIntegrationPackWallet is the complete path to the Oracle wallet in BRM Integration Pack.</td>
</tr>
<tr>
<td>xrefDatabase</td>
<td><code>&lt;xrefDatabase&gt;</code></td>
<td>Contains the details about the cross-reference database, where:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;connectionInfo&gt;</code></td>
<td>■ CrossRefUserName specifies the cross-reference database user name</td>
</tr>
<tr>
<td></td>
<td><code>&lt;login&gt;</code></td>
<td>■ CrossRefHostName specifies the IP address or the host name of the computer on which the cross-reference database is configured</td>
</tr>
<tr>
<td></td>
<td><code>&lt;hostName&gt;</code></td>
<td>■ CrossRefPort specifies the port number assigned to the cross-reference database</td>
</tr>
<tr>
<td></td>
<td><code>&lt;serviceName&gt;</code></td>
<td>■ CrossRefServiceName specifies the name of the cross-reference database service</td>
</tr>
<tr>
<td>pricingServer</td>
<td><code>&lt;pricingServer&gt;</code></td>
<td>Contains the PDC server information, where:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;connectionInfo&gt;</code></td>
<td>■ PricingServerHostName specifies the IP address or the host name of the computer on which PDC is deployed</td>
</tr>
<tr>
<td></td>
<td><code>&lt;adminUser&gt;</code></td>
<td>■ PricingServerPort specifies the port number of the domain on which PDC is deployed</td>
</tr>
<tr>
<td></td>
<td><code>&lt;pdcUser&gt;</code></td>
<td>■ AdminUserName specifies the user name of the PDC server administrator</td>
</tr>
<tr>
<td></td>
<td><code>&lt;adminUser&gt;</code></td>
<td>■ PDCUser specifies the user name of the PDC user</td>
</tr>
<tr>
<td>transformationHome</td>
<td><code>&lt;transformationHome&gt;</code></td>
<td>Specifies the path to the directory that stores the transformation process IDs, where TransformationHome is the complete path to the directory.</td>
</tr>
<tr>
<td>syncPDCLogFileLocation</td>
<td><code>&lt;syncPDCLogFileLocation&gt;</code></td>
<td>Specifies the path to the directory that stores SyncPDC log files, where SyncPDCLogFile is the complete path and the name of the log file.</td>
</tr>
<tr>
<td>brand</td>
<td><code>&lt;brand&gt;</code></td>
<td>Specifies whether PDC supports branding, where BrandOption is:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;BrandOption&gt;</code></td>
<td>■ enabled to specify that PDC supports branding</td>
</tr>
<tr>
<td></td>
<td><code>&lt;disabled&gt;</code></td>
<td>■ disabled to specify that PDC does not support branding</td>
</tr>
<tr>
<td>pdcSSL</td>
<td><code>&lt;pdcSSL&gt;</code></td>
<td>Specifies whether the PDC server supports SSL, where SSLOption is:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;SSLOption&gt;</code></td>
<td>■ enabled to specify that PDC supports SSL. If SSL is enabled, SyncPDC uses the t3s://Host:Port URL to access PDC</td>
</tr>
<tr>
<td></td>
<td><code>&lt;disabled&gt;</code></td>
<td>■ disabled to specify that PDC does not support SSL. If SSL is disabled, SyncPDC uses the t3://Host:Port URL to access PDC</td>
</tr>
<tr>
<td>retryInfo</td>
<td><code>&lt;retryInfo&gt;</code></td>
<td>Contains the information about when and how many times SyncPDC should retry the synchronization process if it fails initially, where:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;retryInterval&gt;</code></td>
<td>■ RetryInterval specifies the interval in seconds after which SyncPDC again tries to synchronize the data that was not previously synchronized</td>
</tr>
<tr>
<td></td>
<td><code>&lt;maxRetries&gt;</code></td>
<td>■ MaxRetries specifies the maximum number of retries</td>
</tr>
</tbody>
</table>
### Table 3–1 (Cont.) Elements in the SyncPDCConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| ECESync                  | `<ECESync>ECESyncOption</ECESync>` | Specifies to synchronize BRM data with PDC, where `ECESyncOption` is:  
  - **true** to synchronize BRM data with PDC to use ECE for usage rating.  
  - **false** to synchronize BRM data with PDC to use real-time and batch rating engines for usage rating. |
| pdcXML                   | `<pdcXML>PDCData</pdcXML>`          | Specifies the path to the directory where SyncPDC creates the XML files containing the extracted data in PDC format, where `PDCData` is the complete path to the directory. |
| reportFileLocation       | `<reportFileLocation>ReportLocation</reportFileLocation>` | Specifies the path to the directory where SyncPDC stores the report of the synchronization process, where `ReportLocation` is the complete path to the directory. |
| brmExtractedXML          | `<brmExtractedXML>BRMData</brmExtractedXML>` | Specifies the path to the directory where SyncPDC creates the XML files containing the data extracted from the BRM database, where `BRMData` is the complete path to the directory. |
| pdcExtractedXML          | `<pdcExtractedXML>PDCData</pdcExtractedXML>` | Specifies the path to the directory where SyncPDC creates the XML files containing the data extracted from the PDC database, where `PDCData` is the complete path to the directory. |
| archiveFileLocation      | `<archiveFileLocation>ArchiveFile</archiveFileLocation>` | Specifies the path to the directory where a successfully processed file is archived, where `ArchiveFile` is the complete path to the directory. No archive file is created if there is no change from the previous synchronization. |
| skipBREMigation          | `<skipBREMigation> SkipOption </skipBREMigation>` | Specifies to skip synchronization of pipeline configuration data, where `SkipOption` is either true or false. |

3-4  PDC Creating Product Offerings
About the SyncPDC Utility

Synchronizing Pricing Setup Components

3-5

### Table 3–1 (Cont.) Elements in the SyncPDCConfiguration.xml File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fieldSelection</td>
<td>&lt;fieldSelection&gt; &lt;targetEngine&gt;TargetRatingEngine&lt;/targetEngine&gt;</td>
<td>Contains the information about the target rating engine and the BRM event fields provided as input to the target rating engine for usage rating.</td>
</tr>
<tr>
<td></td>
<td>&lt;eventName&gt;EventName&lt;/eventName&gt;</td>
<td>&lt;targetEngine&gt; element specifies the target rating engine used for usage rating, where:</td>
</tr>
<tr>
<td></td>
<td>&lt;fullyQualifiedName&gt;EventFieldName&lt;/fullyQualifiedName&gt;</td>
<td>■ <strong>Convergent Charging.</strong> Specifies that ECE is used for usage rating.</td>
</tr>
<tr>
<td></td>
<td>&lt;/eventName&gt;</td>
<td>■ <strong>Realtime Charging.</strong> Specifies that the real-time rating engine is used for usage rating.</td>
</tr>
<tr>
<td></td>
<td>&lt;/fullyQualifiedName&gt;</td>
<td>■ <strong>Batch Charging.</strong> Specifies that the batch rating engine is used for usage rating.</td>
</tr>
<tr>
<td></td>
<td>&lt;/eventFields&gt;</td>
<td>&lt;eventFields&gt; element specifies the fields that are provided as input to TargetRatingEngine, where:</td>
</tr>
<tr>
<td></td>
<td>&lt;eventName&gt;EventName&lt;/eventName&gt;</td>
<td>■ <strong>EventName</strong> specifies the class name of the BRM event; for example,./event.</td>
</tr>
<tr>
<td></td>
<td>&lt;fullyQualifiedName&gt;EventFieldName&lt;/fullyQualifiedName&gt;</td>
<td>■ <strong>EventFieldName</strong> specifies the fully qualified name of the BRM event field that is provided as input to TargetRatingEngine; for example, PIN_FLD_NAME.</td>
</tr>
<tr>
<td></td>
<td>&lt;/eventName&gt;</td>
<td><strong>Note:</strong> Each &lt;fieldSelection&gt; element can have multiple &lt;eventFields&gt; elements but only one &lt;targetEngine&gt; element. To support multiple target rating engines, add the &lt;fieldSelection&gt; element for each target rating engine.</td>
</tr>
<tr>
<td></td>
<td>&lt;/fullyQualifiedName&gt;</td>
<td>Similarly, each &lt;eventFields&gt; element can have multiple &lt;fullyQualifiedName&gt; elements but only one &lt;eventName&gt; element. To rate multiple events, add the &lt;eventFields&gt; element for each event.</td>
</tr>
</tbody>
</table>

---

Synchronizing Pricing Setup Components 3-5
**Table 3–1 (Cont.) Elements in the SyncPDCConfiguration.xml File**

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>syncPDCBREConfig</code></td>
<td><code>&lt;syncPDCBREConfig&gt;</code></td>
<td>Contains the information about EDR container description data and delayed events and services for batch rating engine.</td>
</tr>
<tr>
<td></td>
<td><code>  &lt;containerDesc&gt;</code></td>
<td>The <code>&lt;containerDesc&gt;</code> element specifies the EDR container description data that is used by the batch rating engine, where:</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;param&gt;</code></td>
<td>■ <code>ParamName</code> specifies the field name that defines the EDR container description name</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;paramname&gt;ParamName&lt;/paramname&gt;</code></td>
<td>■ <code>ParamValue</code> specifies the value of <code>ParamName</code>. The EDR container description data that matched <code>ParamValue</code> is used as the base to synchronize EDR fields for events and services</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;/param&gt;</code></td>
<td>● <code>BRMBatchRatingEventName</code> specifies the delayed event name</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;/containerDesc&gt;</code></td>
<td>● <code>EventExtensionBlockName</code> specifies the extension block name to which the delayed event is mapped</td>
</tr>
<tr>
<td></td>
<td><code>  &lt;eventExtension&gt;</code></td>
<td>Add <code>&lt;param&gt;</code> elements to the <code>&lt;eventExtension&gt;</code> elements for each <code>BRMBatchRatingEventName</code> and <code>EventExtensionBlockName</code> to include.</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;param&gt;</code></td>
<td>The <code>&lt;eventExtension&gt;</code> element specifies the mapping of a delayed event to the extension block name used in the batch rating engine, where:</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;paramname&gt;BRMBatchRatingEventName&lt;/paramname&gt;</code></td>
<td>■ <code>BRMBatchRatingEventName</code> specifies the delayed event name</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;/param&gt;</code></td>
<td>● <code>EventExtensionBlockName</code> specifies the extension block name to which the delayed event is mapped</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;/eventExtension&gt;</code></td>
<td>Add <code>&lt;param&gt;</code> elements to the <code>&lt;serviceExtension&gt;</code> elements for each <code>BRMServiceName</code> and <code>ServiceExtensionBlockName</code> to include.</td>
</tr>
<tr>
<td></td>
<td><code>  &lt;serviceExtension&gt;</code></td>
<td>The <code>&lt;serviceExtension&gt;</code> element specifies the mapping of a BRM service to the extension block name used in the batch rating engine, where:</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;param&gt;</code></td>
<td>■ <code>BRMServiceName</code> specifies the BRM service name</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;paramname&gt;BRMServiceName&lt;/paramname&gt;</code></td>
<td>● <code>ServiceExtensionBlockName</code> specifies the extension block name to which the BRM service is mapped</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;/param&gt;</code></td>
<td>Add <code>&lt;param&gt;</code> elements to the <code>&lt;serviceExtension&gt;</code> elements for each <code>BRMServiceName</code> and <code>ServiceExtensionBlockName</code> to include.</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;/serviceExtension&gt;</code></td>
<td>For more information about the EDR container description, see the BRM documentation.</td>
</tr>
<tr>
<td></td>
<td><code>  &lt;/syncPDCBREConfig&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>scheduling</code></td>
<td><code>&lt;scheduling&gt;</code></td>
<td>Contains the scheduling configurations for SyncPDC. If the <code>&lt;scheduling&gt;</code> element is not present or its child elements are not defined correctly, SyncPDC runs immediate only once.</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;runOnStartup&gt;StartupOption&lt;/runOnStartup&gt;</code></td>
<td><code>StartupOption</code> specifies whether SyncPDC should run immediately:</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;startAt&gt;StartTime&lt;/startAt&gt;</code></td>
<td>■ <code>true</code> specifies that SyncPDC runs immediately in addition to the scheduled time</td>
</tr>
<tr>
<td></td>
<td><code>      &lt;interval&gt;Interval&lt;/interval&gt;</code></td>
<td>■ <code>false</code> specifies that SyncPDC does not run immediately but at the scheduled time</td>
</tr>
<tr>
<td></td>
<td><code>    &lt;/scheduling&gt;</code></td>
<td><code>StartTime</code> specifies the time in <code>HH:MM</code> format when you want SyncPDC to start, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>HH</code> specifies hours between 00 and 23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>MM</code> specifies minutes between 00 and 59. <code>Interval</code> specifies the frequency in <code>N:A</code> format at which you want to run SyncPDC, where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>N</code> is an integer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>■ <code>A</code> is <code>D</code> for days, <code>H</code> for hours, or <code>M</code> for minutes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For example, to run SyncPDC every two days, enter: <code>&lt;interval&gt;2:D&lt;/interval&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> <code>24:H</code> is not same as <code>1:D</code> due to daylight saving time.</td>
</tr>
</tbody>
</table>
Updating the SyncPDCConfiguration.xml File

To update SyncPDCConfiguration.xml:

1. Open the BRM_Integration_Pack_Home/apps/syncpdc/SyncPDCConfiguration.xml file in a text editor or an XML editor.
2. Edit the file. See Table 3–1 for more information.
3. Save and close the file.
4. Run the SyncPDC utility.
   See "Running SyncPDC" for more information.

Running SyncPDC

To run SyncPDC:

1. Ensure that transformation engines are running. See the discussion of starting the transformation engines in PDC Installation Guide.
2. Go to the BRM_Integration_Pack_Home/apps/syncpdc directory.
3. Enter the following command:
   `startSyncPDC`
   The Enter Wallet Password prompt appears.
4. Enter the BRM Integration Pack wallet password.
   A series of messages appears on the command prompt that indicate the synchronization status.
   For example:
   
   Clean up work item SYNC_EVENT...
   Work item SYNC_EVENT started (item 1 of 8).
   Processing EXTRACT work action...
   Work action EXTRACT completed.
   Processing ANALYZE work action...
   Work action ANALYZE completed.
   Processing TRANSFORM work action...
   Work action TRANSFORM completed.
   Processing LOAD work action...
   Work action LOAD completed.
   Work item SYNC_EVENT completed.

   After the synchronization is complete, check the synchronization report available in the directory defined in the <reportFileLocation> element of the SyncPDCConfiguration.xml file.

Scheduling Synchronization

You can schedule the data synchronization from BRM or rating systems with PDC by:

- Running SyncPDC Immediately
- Running SyncPDC at a Scheduled Time
- Running SyncPDC Immediately and at a Scheduled Time
- Running SyncPDC at Recurring Scheduled Time
Running SyncPDC Immediately
To run SyncPDC immediately instead of waiting for the next scheduled run time:

1. Open the BRM_Integration_Pack_Home/apps/syncpdc/SyncPDCConfiguration.xml file in a text editor or an XML editor.
2. Set the <runOnStartup> element to true:
   <runOnStartup>true</runOnStartup>
3. Save and close the file.
4. (Optional) If SyncPDC is already running in the background, stop it.
   See "Stopping SyncPDC" for more information.
5. Run SyncPDC. See "Running SyncPDC" for more information.

Running SyncPDC at a Scheduled Time
To run SyncPDC at a schedule time:

1. Open the BRM_Integration_Pack_Home/apps/syncpdc/SyncPDCConfiguration.xml file in a text editor or an XML editor.
2. Set the <runOnStartup> element to false:
   <runOnStartup>false</runOnStartup>
3. Set the <startAt> element to the start time:
   <startAt>HH:MM</startAt>
   where:
   ■ HH specifies hours between 00 and 23
   ■ MM specifies minutes between 00 and 59
   For example, to run SyncPDC at 11:50 pm, enter:
   <startAt>23:50</startAt>
4. Save and close the file.
5. Run SyncPDC.
   See "Running SyncPDC" for more information.

Running SyncPDC Immediately and at a Scheduled Time
To run SyncPDC immediately and at a schedule time:

1. Open the BRM_Integration_Pack_Home/apps/syncpdc/SyncPDCConfiguration.xml file in a text editor or an XML editor.
2. Set the <runOnStartup> element to true:
   <runOnStartup>true</runOnStartup>
3. Set the <startAt> element to the start time:
   <startAt>HH:MM</startAt>
   where:
   ■ HH specifies hours between 00 and 23

3-8  PDC Creating Product Offerings
About the SyncPDC Utility

Synchronizing Pricing Setup Components

3-9

■ MM specifies minutes between 00 and 59

For example, to run SyncPDC immediately and at 11:50 pm, enter:

<runOnStartup>true</runOnStartup>
<startAt>23:50</startAt>

4. Save and close the file.

5. Run SyncPDC.

See "Running SyncPDC" for more information.

Running SyncPDC at Recurring Scheduled Time

To run SyncPDC at recurring scheduled time:

1. Open the BRM_Integration_Pack_Home/apps/syncpdc/SyncPDCCConfiguration.xml file in a text editor or an XML editor.

2. Set the <runOnStartup> element to false:

   <runOnStartup>false</runOnStartup>

3. Set the <startAt> element to the start time:

   <startAt>HH:MM</startAt>

   where:

   ■ HH specifies hours between 00 and 23
   ■ MM specifies minutes between 00 and 59

4. Specify the <interval> element as follows:

   <interval>N:A</interval>

   where:

   ■ N is an integer
   ■ A is D for days, H for hours, or M for minutes

For example, to run SyncPDC at 11:50 pm everyday, enter:

<runOnStartup>false</runOnStartup>
<startAt>23:50</startAt>
<interval>1:D</interval>

5. Save and close the file.

6. Run SyncPDC.

See "Running SyncPDC" for more information.

Synchronizing BRM Data with PDC to Use ECE for Usage Rating

To synchronize BRM data with PDC to use ECE for usage rating:

1. Open the BRM_Integration_Pack_Home/apps/syncpdc/SyncPDCCConfiguration.xml file in a text editor or an XML editor.

2. Set the <ECESync> element to true:

   <ECESync>true</ECESync>
3. Save and close the file.

4. Run SyncPDC.

   See “Running SyncPDC” for more information.

**Stopping SyncPDC**

You can stop SyncPDC at any time. If you stop SyncPDC while it is synchronizing data, it finishes the synchronization before stopping.

To stop SyncPDC, enter the following command:

```
stopSyncPDC
```

A series of messages appears, indicating progress.

For example:

```
Previous process id :4381
Killing child processes ... 
Killing process ... 
Cleaning temporary files ...
[1] + Killed
```

**Generating Synchronization Reports**

SyncPDC generates reports for the components that it synchronizes. If there are no changes from the previous synchronization, SyncPDC does not create a report.

You specify the location of the SyncPDC report in the `<reportFileLocation>` element of the SyncPDCConfiguration.xml file.

When synchronizing data from BRM with PDC, the SyncPDC report lists the object type for each setup component that is synchronized. It lists the PDC display name and the corresponding BRM or rating system name for each of the setup component objects. It also provides the total number of object types for each setup component that are synchronized.

Figure 3–1 shows a sample SyncPDC report for BRM data.

---

**Figure 3–1 Sample SyncPDC Report for BRM Data**

<table>
<thead>
<tr>
<th>PDC Internal Id</th>
<th>PDC Event Name</th>
<th>BRM Event Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>df72f756-ed1c-4be3-93df-ea4737e1a09</td>
<td>Event</td>
<td>event</td>
</tr>
<tr>
<td>9a34af04-98b6-4c5f-869b-68bc4e6c7f40</td>
<td>EventActivityContent</td>
<td>eventactivitycontent</td>
</tr>
<tr>
<td>2c5df822-adfa-4920-bb0b-e77f53e6d954</td>
<td>EventActivitySettlement</td>
<td>eventactivitysettlement</td>
</tr>
<tr>
<td>b3143f96-3c4c-4564-85ea-15f54a21b9ed</td>
<td>EventActivityToke</td>
<td>eventactivitytoke</td>
</tr>
<tr>
<td>51a77229-ebab-1e6f-90de-9a7f7f890a8d</td>
<td>EventBillingCycleDiscount</td>
<td>eventbillingcycledonoted</td>
</tr>
<tr>
<td>057618-54cc-4b4a-8e3c-754ac448ead</td>
<td>EventBillingCycleDiscountNotCalled</td>
<td>eventbillingcycledonotcalled</td>
</tr>
<tr>
<td>b65c3917-3053-8b43-bb13-b9f6f586081c</td>
<td>EventBillingCycleFold</td>
<td>eventbillingcyclofold</td>
</tr>
<tr>
<td>b2605d4f-69fe-442a-96b6-4f6e6595d6d7</td>
<td>EventBillingCycleRollerMonthly</td>
<td>eventbillingcyclerollermonthly</td>
</tr>
<tr>
<td>51b04x2-7101-4b47-8a54-1a122e48181</td>
<td>EventBillingFeeFaledPayment</td>
<td>eventbillingfeefailedpayment</td>
</tr>
<tr>
<td>c37e996-0202-7ee-0f5a-043f3d669da5</td>
<td>EventBillingIncentive</td>
<td>eventbillingincentive</td>
</tr>
<tr>
<td>f23add0-3093-4a60-957a-a397f92e38b6</td>
<td>EventBillingProductFeeCancel</td>
<td>eventbillingproductfee cancels</td>
</tr>
<tr>
<td>3badd520-2c95-884e-b26-78bcf3e0d9f</td>
<td>EventBillingProductFeeCycleCycled</td>
<td>eventbillingproductfeecyclecycled</td>
</tr>
<tr>
<td>9d79dca-8f2a-4e16-9ce-68c12d1939a</td>
<td>EventBillingProductFeeCycleForwardAnnual</td>
<td>eventbillingproductfeecycleforwardannual</td>
</tr>
</tbody>
</table>

Total number of BRM event objects synchronized: 13
When synchronizing data from ECE with PDC, the SyncPDC report provides the status of loading the XML files containing ECE data.

Figure 3–2 shows a sample SyncPDC report for ECE data.

**Figure 3–2  Sample SyncPDC Report for ECE Data**

<table>
<thead>
<tr>
<th>Import File Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>PdcService.xml</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>Customer.xml</td>
<td>SUCCESS</td>
</tr>
<tr>
<td>PdcEvent.xml</td>
<td>FAILED</td>
</tr>
</tbody>
</table>

Successfully Imported Files: 2
Failed to Import Files: 1
End Load ECE Report.

**Handling Synchronization Failures**

The most common reasons for a synchronization failure are as follows:

- The billing or rating system is down
- After billing or rating system data was modified, the billing or rating system was not restarted. Hence, no data is available for synchronization with PDC
- The PDC system is down
- The transformation engines are down
- The transformation engines failed to process the PDC data and update the cross-reference database
- The billing or rating system database or the PDC database is down

**Reprocessing a Failed Synchronization**

To reprocess a failed synchronization:

1. Identify and fix the cause of the synchronization failure.
   
   For information about synchronization failures, check the SyncPDC log file located in the directory specified in the `<syncPDCLogFileLocation>` element of the SyncPDCConfiguration.xml file.

2. Run SyncPDC.
   
   See "Running SyncPDC" for more information.

   SyncPDC synchronizes the data that failed to synchronize earlier and ignores the data that was already synchronized.
Changing Display Names

When synchronizing setup components, SyncPDC creates display names in PDC for the corresponding data objects and their fields. You can change the default display names that SyncPDC creates.

To change the display name of a setup component:

1. Export the setup component to an XML file by using the ImportExportPricing utility.
   See "Exporting Pricing and Setup Components from PDC" for more information.
2. Open the XML file in a text editor or an XML editor.
3. Change the display name of the setup component in the XML file.
4. Save and close the file.
5. Import the updated XML file to PDC by using ImportExportPricing.
   See "Importing Pricing and Setup Components" for more information.

If you later run SyncPDC to synchronize any changes from the billing or rating system with PDC, SyncPDC retains the updated component name.
About Charge Offers and Charges

Charge offers determine the price of one or more events associated with a service. For example:

- For a mobile phone service:
  - 10 cents per call on peak time
  - 5 cents per call on off-peak time
  - $20 setup fee

- For a video download service:
  - $10 monthly fee
  - $1 fee per video download

A charge offer includes one or more charges. Figure 4-1 shows a charge offer in PDC. This charge offer includes two charges:

- ConvergentUsage charges 10 cents per minute for voice usage.
- Monthly Recurring Charge charges a 20-dollar monthly fee, and credits 100 minutes per month.

**Figure 4–1  Charge Offer in PDC**

Each charge can have one or more balance impacts. In Figure 4–1 the balance impact for usage is 10 cents per minute. The charge is determined primarily by the following columns in the balance impact in PDC:

- **Impact:** Can be a debit or a credit.
- **Amount:** The amount of the balance impact.
- **Balance Element:** The type of balance that is impacted. For a debit impact, this is usually a currency balance. For a credit impact, this is usually a noncurrency balance.
- **Per Unit:** What the charge is based on; for example, number of megabytes downloaded or number of minutes used.

The **Per Unit** value is typically the RUM that the charge is based on, or a measurement related to it. For example, if the RUM used to rate the event is seconds, the **Per Unit** value can be seconds, minutes, or hours.

For usage charges, you can also use the following options for the **Per Unit** value:

- **Fixed Charge:** Use this option to charge a flat fee for the event. Fixed charges are unaffected by the quantity of the event.
- **Amount Used:** This bases the charge on the quantity of the event is measured. For example, to charge for SMS messages, each event is one message, so **Amount Used** is always one message. If the **Amount** column is .10 and the **Per Unit** column is **Amount Used**, the charge is 10 cents for each SMS message.
Figure 4–2 shows a balance impact that charges for the amount used. This balance impact could be used for charging an SMS service, a video download, and so on.

**Figure 4–2  Balance Impact With No RUM Specified**

<table>
<thead>
<tr>
<th>Impact</th>
<th>Amount Balance Element</th>
<th>Per Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debit</td>
<td>0.50 US Dollar</td>
<td>Amount Used</td>
</tr>
</tbody>
</table>

The options available in a balance impact depend on the charge offer type and the charge type. For example, the Per Unit value is not used for one-time charges such as purchase fees, or for recurring charges. In those cases, the event is always charged the same amount.

Figure 4–3 shows the balance impacts for a charge that charges a recurring $20 monthly fee, and credits 500 minutes every month.

**Figure 4–3  Balance Impacts for Monthly Fee and Monthly Megabyte Credit**

<table>
<thead>
<tr>
<th>Impact</th>
<th>Amount Balance Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debit</td>
<td>20.00 US Dollar</td>
</tr>
<tr>
<td>Credit</td>
<td>500.00 Included Megabytes</td>
</tr>
</tbody>
</table>

When you create a charge offer, you do the following:

1. Create the charge offer in PDC. In the charge offer, you define charge offer attributes such as the service that the charges apply to, how a customer can purchase and own the charge offer, tax settings, and so on. See "Configuring Charge Offer Usage and Ownership."

2. Create one or more charges. For each charge, you define the type of charge; for example, a monthly fee or a usage fee, how to measure the charge, and other properties.

3. Define the pricing for each charge. The pricing specifies one or more balance impacts, and how to apply the balance impacts; for example, based on the quantity of the event.

**Configuring Charge Offer Usage and Ownership**

When you create a charge offer, you associate it with one of the following:

- A single service. For example, to charge for GSM usage, associate the charge offer with the GSM telco service.
  
  Multiple charge offers can be associated with the same service.

- No service. In this case, you associate the charge offer with Account. You might create a charge offer for late charges or for coupons.
  
  You cannot charge for usage events with a charge offer set to Account.

When you create a charge offer, you specify the charge offer type and the settings that control how the charge offer can be purchased and owned. The charge offer type determines how the charge offer can be used:
Subscription: These charge offers are purchased and owned by subscribers, and the charges apply only to the charge offer owner. Most charge offers are subscription charge offers. They can charge for any event type.

Item: These charge offers contain charges that are applied only once, for example, a purchase fee or cancel fee. Item charge offers cannot include recurring charges or usage changes.

System: These charge offers apply to all subscribers who use a particular service. System charge offers are not owned by subscribers. They cannot contain usage charges.

In addition to defining the service and the charge offer type, you can define the following settings:

- When the charge offer can be purchased; which can be immediately, or in a specified date range.

  **Note:** To be added to a bundle, a charge offer must have a purchase period that is the same as or greater than the bundle’s purchase period.

- The validity period for the charge offer. For example, the validity period specifies when the customer can use the service that the charge offer applies to. The validity period is not set in the charge offer itself. It is set when you configure the bundle that the charge offer belongs to. See "About the Validity Periods of Offers in Bundles"

- The charge offer priority. When multiple charge offers apply to the same service-event pair, BRM can consider the charge offers in the order of their priority. If a portion of the event remains unrated after BRM applies all the charges in the charge offer with the highest priority, each subsequent priority charge offer is analyzed until the entire event is rated.

- How the charge offer can be owned or purchased.
  - **Owned:** For example, if a charge offer provides an email service, you might want to limit the number of email login names that a customer can own.
  - **Purchased:** For example, if a charge offer includes an item such as a t-shirt, you might want to limit the number of t-shirts that can be purchased simultaneously.
  - **Partially purchased:** For example, if a charge offer gives customers 300 minutes of off-peak calls for $30, customers might be permitted to purchase half that amount for half the price.

Figure 4–1 shows the charge offer settings in PDC.
Restricting the End Time of Granted Balances That Start on First Usage

You can configure BRM to automatically restrict the validity period of granted noncurrency balances to end no later than the end time of the charge offer or discount offer that grants the balance.

When you configure balance validity to start on first usage and end on a date relative to the start date (when first usage occurs), you cannot know the actual end date when setting up the offer. Restricting the balance end time to the offer’s end time ensures that the balance cannot continue to be consumed after the offer expires.

---

**Note:** When an offer is cancelled, the validity period end time of balances granted by that charge offer or discount offer is set to the time of the cancellation.

---

When you restrict balance validity end time, BRM sets the end time to the offer’s end time at the time of the grant. When the customer consumes the granted balance for the first time, the relative end time is calculated:

- If the calculated end time is later than the granting offer’s end time, the balance validity period uses the offer’s end time.
- If the calculated end time is earlier than the granting offer’s end time, the balance validity period uses the calculated end time.
To enable this feature, run the `pin_bus_params` utility to change the `RestrictResourceValidityToOffer` business parameter. For information about this utility, see *BRM Developer’s Guide*.

To restrict balance validity end times to charge offer or discount offer end times:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```bash
   pin_bus_params -r BusParamsMultiBal bus_params_multi_bal.xml
   ``
3. In the file, change `FALSE` to `TRUE`:
   ```xml
   <RestrictResourceValidityToOffer>FALSE</RestrictResourceValidityToOffer>
   ``
4. Save the file as `bus_params_multi_bal.xml`.
5. Load the XML file into the BRM database:
   ```bash
   pin_bus_params bus_params_multi_bal.xml
   ``
6. Stop and restart the CM.
7. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see *BRM System Administrator’s Guide*.

**Charge Offers and Tax Calculation**

You can use the following tax calculation settings:

- In a charge offer, you can specify a tax supplier.
- In a charge, you can specify at tax code to use for the charge.
- In a charge, you can specify when to apply taxes, when the charge is calculated, or during billing.

You need to define tax codes and tax suppliers in BRM before you can assign them in discounts. For information, see *BRM Calculating Taxes*.

**Charge Offers and General Ledger**

You assign a general ledger ID (G/L ID) to balance impacts. This enables you to track revenue for each type of charge; for example, you can assign different G/L IDs for monthly fees and usage charges.
This document describes how to configure charges in Oracle Communications Billing and Revenue Management (BRM) charge offers.

Topics in this document:

- About Charges
- Specifying When a Charge is Effective
- Applying a Charge to Inactive or Canceled Charge Offers
- Configuring In-Advance Billing in Charges
- Configuring Cycle Alignment for Recurring Charges
- About Prorating Recurring Charges and Rollovers

See also:

- About Creating Product Offerings
- Configuring Charge Offers
- Configuring Pricing in Charge Offers
- Pricing Configurations in Charge Offers
- Creating Discount Offers

About Charges

Each charge in a charge offer has a charge category and a charge type. The charge category determines which events the charge can be configured for:

- **Usage events**: Charges for the use of a service, such as telephone calls or broadband sessions.
- **Recurring events**: Ongoing charges that are not generated or affected by usage, such as a monthly subscription fee.
- **One-time events**: Nonrecurring charges, such as setup or cancellation fees.
- **Rollover events**: Charges that extend the validity of unused balances to succeeding cycles. For example, included minutes are often rolled over.
- **Fold events**: Charges used to zero-out a balance or convert one balance into another. For example, you could configure a fold charge to zero-out unused included hours at the end of each month or convert frequent flyer miles to a dollar amount. Fold charges must be associated with an existing charge selector whose event matches the fold event (see "About Selectors").
The charge type depends on the charge category. For example:

- If the charge category is **Recurring**, the charge type could be monthly or yearly.
- If the charge category is **Usage**, the charge type could be voice usage or data download.

In addition to the charge category and charge type, you define the following:

- The RUM to use to measure the event; for example, **Duration** or **Messages**. See "Configuring Ratable Usage Metrics (RUMs)" for more information.
- The currency specified for a charge, such as U.S. dollars or euro. This currency defines the currency balance element that you can select when configuring pricing for the charge.
- The dates during which the charge is effective. By default, this period starts immediately and never ends. See "Specifying When a Charge is Effective" for more information.
- Whether to apply the charge if the account is inactive or canceled. See "Applying a Charge to Inactive or Canceled Charge Offers" for more information.
- Settings for recurring charges and rollovers. See:
  - "Configuring In-Advance Billing in Charges"
  - Configuring Cycle Alignment for Recurring Charges
  - About Prorating Recurring Charges and Rollovers

**Specifying When a Charge is Effective**

You can apply a date range to a charge; for example, set a fixed start date and an end date; or specify a date range relative to when the charge offer was purchased. If you include multiple charges in a charge offer, you can configure multiple date ranges for each charge.

If you want to apply the same charge with multiple variations applied in different date ranges, you can use multiple date ranges in a single charge. You can use different pricing for each date range. For example, you could create a charge that charges $10 for the first date range, and $30 for the second date range.

BRM supports the following types of charge date ranges:

- **Fixed**: Specifies a period that starts and ends on particular dates. For example:
  - Immediately through 6/1/2012
  - 6/1/2012 through 1/1/2013
  - 1/1/2013 through never ends

  Fixed date ranges cannot overlap.

- **Relative**: Begins at a time relative to the time the charge is purchased and continues for a specified length of time, such as days, hours, minutes, or seconds. The purchase date is the day the charge offer is added to the account. Unlike fixed date ranges, relative date ranges can overlap.

**Note:** In addition to specifying date ranges for charges, you can specify date ranges for the balance impacts in a charge. See "Configuring Effective Dates for Pricing" for information.
Applying a Charge to Inactive or Canceled Charge Offers

For each charge, you specify whether to stop charging if the charge offer is inactivated or canceled. The options are:

- Don’t stop charging if the charge offer is inactive or canceled
- Stop charging if the charge offer is inactivated
- Stop charging if the charge offer is canceled
- Stop charging if the charge offer is inactivated or canceled

Configuring In-Advance Billing in Charges

In-advance billing enables you to charge customers in advance in the first bill. For each charge, you can configure how far in advance to bill the customer. For example, if a customer purchases an offer on May 1 and the offer’s $10 monthly fee is billed three months in advance, the total charge in the customer’s first bill is $30. The total charge in the next (June) bill is $10, but from an accounting perspective, the $10 fee applies to August, not to June.

Configuring Cycle Alignment for Recurring Charges

Recurring charges are based on BRM system events that occur on cycles. You can apply recurring charges in either or the following ways:

The following options apply only to recurring charges:

- On the charge offer purchase date. For example, if the billing date is the 1st of the month and the charge offer is purchased on January 10, the charge is applied on the 10th of every month (for the interval January 10 to February 10, February 10 to March 10, and so on).

  On the customer’s current billing date (default). Using the previous example, the charge is prorated and applied on January 10 for the interval January 10 to February 1. For subsequent cycles, the charge is applied on the billing date (for the interval February 1 to March 1, March 1 to April 1, and so on).

About Prorating Recurring Charges and Rollovers

Customers can purchase charge offers and cancel charge offers in the middle of a billing cycle. When you set up recurring charges and rollovers in PDC, you can specify whether the charge or rollover amounts are prorated for the first and last billing cycles based on the number of days in the cycles that the charge offer is owned.

For recurring charges, you can specify whether to charge the full amount, prorate the charge, or apply no charge.

For rollovers, you can specify whether to roll over the entire amount, prorate the rollover amount, or not roll over any of the available balance.
This document describes how to configure pricing in Oracle Communications Billing and Revenue Management (BRM) charge offers.

Topics in this document:

- Configuring Pricing in Charges
- Configuring Validity Periods for Noncurrency Credit Balances
- Specifying Rounding in Balance Impacts
- Specifying if a Balance Impact is Discountable
- About Minimum Charges
- About Conditional Balance Impacts
- About Prorating Charges

See also:

- About Creating Product Offerings
- Configuring Charge Offers
- Configuring Charges in Charge Offers
- Pricing Configurations in Charge Offers
- Creating Discount Offers

**Configuring Pricing in Charges**

Each charge can contain one or more balance impacts. You configure balance impacts in the pricing table. Figure 6–1 shows an example of the pricing table. Each row in the table represents a balance impact.
A balance impact is defined by how much to charge (the **Amount** column in the balance impact) for a given amount (the **Per Unit** column in the balance impact). When configuring pricing for a charge, you can also do the following:

- Configure when to grant noncurrency balances. See "Configuring Validity Periods for Noncurrency Credit Balances."
- Configure how to round balance impacts. See "Specifying Rounding in Balance Impacts."
- Specify if a balance impact can be discounted. See "Specifying if a Balance Impact is Discountable."
- Specify a minimum amount to charge. See "About Minimum Charges."
- Disable proration for a charge if proration is enabled for a charge offer. See "About Prorating Charges."
- Use a price selector to choose a pricing configuration based on the values of specified event, service, or account attributes. See "About Selectors."

### Configuring Validity Periods for Noncurrency Credit Balances

When you create an offer that credits noncurrency balances, such as granting minutes, you can configure a validity period for the balance impact. This is useful if you want to limit how long a customer can keep minutes in their balance before they need to purchase more minutes. **Figure 6–2** shows a balance impact that grants 100 minutes that are available for one billing cycle.

**Figure 6–2 Balance Impact for Minutes**

<table>
<thead>
<tr>
<th>Impact</th>
<th>Amount</th>
<th>Balance Element</th>
<th>Amount Is Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>100.00</td>
<td>Included Minutes ▼</td>
<td>From first usage - 1 Billing Cycles From Validity Start</td>
</tr>
</tbody>
</table>

You can specify the following fixed validity periods:

- For the current accounting cycle
- For the current billing cycle
- For the current recurring cycle
You can also define validity period start and end times. You can start a validity period at these times:

- When the event occurs.
- At first usage of the service.

If a balance is shared among accounts in a sharing group, the balance validity period is set when any account in the group first impacts the balance. Because the same balance is shared with all accounts in the group, the validity period of that balance applies to all accounts.

- From a specified date.
- Relative to a specified date.

You can end a validity period based on:

- A date relative to the start date (measure in minutes, hours, days, months, or in account, billing, or recurring cycles)
- On a specified date.
- No end date.

If a charge offer or discount offer is canceled after being used, the validity end time of any balance granted by the offer is set to the time of the cancellation.

---

**Note:** A charge offer that is valid from first usage should not grant a credit balance, such as included minutes, that is also valid from first usage. If it does, the granted balance cannot be consumed by the first usage event. See “About Granting First-Usage Balance Impacts In First-Usage Charges.”

---

**Specifying Rounding in Balance Impacts**

In a balance impact, you can specify the number of units to which the charge applies. For example, for a mobile phone service, you might charge 0.40 US dollars for every minute of usage in 2-minute increments. In this case, if the unit is Minute, the Increment is 2.

If you specify an increment, you can select a rounding option. Rounding is based on the specified increment. For example, if a 3-minute-and-50-second call is rated in 2-minute increments, it is handled as follows:

- No Rounding: Not applicable. (Applies only to an increment of 1.)
- Round Down: Charge for 2 minutes ($0.80).
- Round Up: Charge for 4 minutes ($1.60).

**Specifying if a Balance Impact is Discountable**

When creating a balance impact, you can specify if it is discountable in a bundle. By default, balance impacts are discountable.

When a charge type is discounted in a bundle, the discount applies to all instances of the charge type that are discountable.

For example, you might have a charge offer that charges for two cycle events. You charge a monthly fee for one event, and you grant included minutes for the other
event. If you specify a percentage discount for cycle charges in the bundle, the
discount applies to both cycle charges.

To discount only the monthly fee, you must make the balance impact for the monthly
fee discountable and the balance impact that grants included minutes
nondiscountable.

About Minimum Charges

You can set a minimum charge for each balance element impacted by a pricing
instance. Minimum charges are configured per price tier (see "About Defining Pricing
Based on Multiple RUMs") and effective period.

If you define different minimum charges in different price tiers for the same balance
element, the largest minimum charge is used during rating. For example, if one step
defines a minimum charge of $2 and another a minimum charge of $1, $2 is used.

When multiple price tiers are applied to a charge, the minimum price is applied if the
total of the charges does not meet the minimum charge. It is not applied for each of the
price tiers if they each do not meet the minimum charge.

About Conditional Balance Impacts

A conditional balance impact is a balance impact that credits or debits a customer’s
balance only when the customer uses a charge offer for the first time within a specified
period; for example, the first time in a day or the first time in two days. For example,
you can use conditional balance impacts to grant daily included minutes to a
customer, instead of using recurring events.

You can align the start time of the conditional balance impact period with the start
time of the associated charge offer, the start of a calendar day, or the event occurrence.
You can also specify whether the balance is available from the start time of the period
or from the time the event occurs.

Conditional balance impacts can only be used with charge offers that are associated
with a service, not with the charge offers associated with an account.

The following example shows how to configure conditional balance impacts for
granting daily included minutes.

In this example:

■ The customer is granted 40 included minutes at $5 per day.

■ The included minutes are valid for a day.

■ The customer is not charged on the days that the offer is not used.

To configure conditional balance impacts for granting daily included minutes:

1. Create a RUM of type Conditional for the desired service and usage event. See the
discussion about creating a new RUM configuration in the PDC Help.

Figure 6–3 shows a RUM of type Conditional.
2. Create a charge offer. See the discussion about creating a charge offer in the PDC Help.

3. Add a new charge to the charge offer. When you add the charge, ensure that you select the following values:
   - **Charge Category**: Usage
   - **Pricing Profile**: Convergent Usage
   - **Measured By**: The RUM that you created in step 1. Optionally, you can select additional RUMs.
   See the discussion about adding a new charge in the PDC Help.

4. Configure conditional balance impacts to grant 40 minutes at $5 per day on the days the charge offer is used.
   Figure 6–4 shows the conditions configured for granting daily included minutes.

   **Figure 6–4 Conditions for Granting Daily Included Minutes**

   ![Balance Impact Condition](image)

   Figure 6–5 shows the validity period for the conditional balance impact.

   **Figure 6–5 Validity Period for a Conditional Balance Impact**

   ![Amount Is Valid](image)

   Figure 6–6 shows the conditional balance impacts used in this example for granting daily included minutes.
About Prorating Charges

You can specify to not prorate a recurring charge when proration is enabled in the charge offer itself. The proration settings are specified in the charge offer; this option only enables or disables proration for each recurring charge.

Configuring Dynamic Pricing

Dynamic pricing allows you to override the price specified in the product offerings at run time.

To override the price, you create a pricing XML file with dynamic tags and import the file into the PDC database by using the `ImportExportPricing` utility.

Dynamic tags are the XML elements that are used for overriding the value of the pricing attributes. ECE uses these attributes to determine the price when processing a usage request.

---

**Important:** Ensure that you create a unique dynamic tag across multiple charge offers since the tags are not scoped to charge offers at run time.

---

Dynamic tags are used for different pricing within a rate plan. PDC provides a sample XML file for dynamic tags in the `PDC_home/apps/Samples/Examples` directory, where `PDC_home` is the directory in which you installed PDC. The dynamic tag (`priceTag`) consists of the following fields:

- **attributeName.** The name of the attribute. The value of the attribute is dynamically determined during rating.
- **tagName.** The unique identifier of the dynamic tag.
- **tagScope.** The scope of the dynamic tag. Following are the available tag scopes:
  - EVENT_PROFILE
  - PRODUCT_PROFILE
  - CUSTOMER_PROFILE

**Note:** Only the EVENT_PROFILE tag scope is supported for charges with the Convergent Usage pricing profile.

- **Description.** (Optional) The description of the dynamic tag.
For example:

```xml
<fixedCharge>
  <price>20.0</price>
  <unitOfMeasure>NONE</unitOfMeasure>
  <balanceElementNumCode>840</balanceElementNumCode>
  <discountable>true</discountable>
  <priceType>CONSUMPTION</priceType>
  <priceTag>
    <attributeName>price</attributeName>
    <tagName>PRICE_TAG_FIXED_PRICE</tagName>
    <tagScope>EVENT_PROFILE</tagScope>
  </priceTag>
</fixedCharge>
```

In this example, the default price is 20.0. This is used by default if an overriding price is not specified.

For information on importing the pricing XML file into the PDC database, see "Importing and Exporting Pricing and Setup Components".

For information on enabling ECE to override the default value of the pricing attributes at run time, see the discussion about configuring ECE to override a product price in *BRM Elastic Charging Engine Implementation Guide*. 


Pricing Configurations in Charge Offers

This document describes various pricing configurations you can use in Oracle Communications Billing and Revenue Management (BRM) charge offers using Pricing Design Center (PDC).

The topics in this document describe how to:

- Calculate a charge based on the amount of usage or frequency of occurrence. See "Configuring Pricing Based on Quantity."

- Create multiple effective periods for pricing. This enables you to implement future price changes for a charge without creating multiple versions of the charge. See "Configuring Effective Dates for Pricing."

- Use multiple RUMs to charge for an event. For example, you can charge on both duration and occurrence. See "About Defining Pricing Based on Multiple RUMs."

- Use price overrides to base a charge on a pre-defined balance impact. See "About Price Overrides."

- Use fold charges to convert one balance to another. See "About Fold Charge Pricing."

- Grant balances during rating. See "Granting First Usage Balances During Rating."

See also:

- About Creating Product Offerings
- Configuring Charge Offers
- Configuring Charges in Charge Offers
- Configuring Pricing in Charge Offers
- Pricing Configurations in Charge Offers

Configuring Pricing Based on Quantity

To calculate a charge based on the amount of usage or frequency of occurrence, you can create quantity ranges in charge pricing. For example, pricing for a telephony service might contain the following quantity ranges based on the total duration of calls during a month:

- No Minimum to 500 minutes: 10 cents per minute
- 500 to 1000 minutes: 5 cents per minute
- 1000 minutes to No Maximum: 1 cent per minute
To define pricing based on quantity, you add quantity ranges and configure different balance impacts for each range.

In addition to specifying what to base the quantity on (usage, charges, and so on), you specify how to apply the charge to the quantity ranges. There are two options:

- **Pick the quantity range containing the value**: Selects a single quantity range that contains the value of the quantity range expression. For example, if the quantity is 70:
  - 0-60 does not apply
  - 60-100 applies
  - 100-Unlimited does not apply

- **Distribute value across applicable quantity ranges**: Selects one or more quantity ranges that overlap the value of the quantity range expression. For example, if the quantity is 70:
  - 0-60 applies
  - 60-100 applies
  - 100-Unlimited does not apply

When you use quantity ranges in combination with time ranges, you can specify if the quantity consumed is relevant to a new time range. See "Rating Events Split across Time Periods" for information.

### Configuring Effective Dates for Pricing

You can apply a date range to a pricing configuration; for example, set a fixed start date and an end date. You can include multiple date ranges in a pricing configuration. This enables you to implement future price changes for a charge without creating multiple versions of the charge. For example, you could create a charge that has a $10 recurring charge valid for one month, and $20 after that.

Date ranges cannot overlap; for example:

- Immediately through 6/1/2015
- 6/1/2015 through 1/1/2016
- 1/1/2016 through never ends

### About Defining Pricing Based on Multiple RUMs

To configure pricing for multiple RUMs within a charge, you use multiple price tiers, one for each RUM. Figure 7–1 shows a charge that contains two price tiers, Volume and Duration.
About Price Overrides

A price override enables you to replace the balance impact of a record imported from another system or to adjust the balance impact by a specified percentage or a fixed amount. For example, you might charge for content downloads that have already been rated by another service provider. A price override enables you to charge a different amount.

About Fold Charge Pricing

A fold charge is used to zero-out a balance, such as unused minutes or to convert one balance to another.

A fold is configured using a charge selector. The event for the charge selector must be the Fold event. After you have configured the charge selector, select it in the charge offer.

The pricing in a fold charge should have a debit for the balance that you are converting and one or more credits for the balances that you are converting to. Figure 7–2 shows the pricing for a fold that converts Frequent Flyer Miles balance to US Dollar balance, $1 for each frequent flyer mile.

See "About Folds" for information about how fold balances are managed.
Granting First Usage Balances During Rating

ECE can grant first-usage balance impacts during rating. For example, you could create a charge offer that includes these balance impacts:

- Debit 5 cents per minute if there are no included minutes.
- Credit 1 minute for each minute paid at 5 cents per minute. These minutes are valid on first usage.

In this example, the charges occur as follows:

1. A subscriber has used up all their included minutes, and is being charged 5 cents per minute.
2. After ten minutes, the subscriber terminates the call. The subscriber is granted 10 minutes.
3. The next call that the subscriber makes uses the ten minutes, granted as first-usage balance impacts.

You can also grant first-usage balance impacts by using a discount. For example, you could create:

- A charge offer that charges 5 cents per minute.
- A discount that credits one SMS message for each called minute.

In this example, the charges occur as follows:

1. A subscriber makes a 10-minute call.
2. The subscriber terminates the call. The subscriber is granted 10 SMS messages, valid at first usage, with a validity end date after 30 days.
Creating Discount Offers

This document describes how to create discount offers in Pricing Design Center (PDC) for Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

■ About Discount Offers and Discounts
■ Configuring How Discount Offers are Purchased and Applied
■ Applying Discounts Based on Quantity
■ About Discount Filters
■ About Discount Triggers
■ About Using Temporary Balances in Discounts
■ Using Functions to Calculate a Balance Impact
■ Applying Multiple Discount Versions to an Event
■ About Billing-Time Discounts
■ Configuring Effective Dates for Discounts
■ Configuring Discount Validity
■ Discount Offers and Tax Calculation
■ Discount Offers and General Ledger
■ About Snowball Discounts

See also:

■ About Creating Product Offerings
■ Configuring Charge Offers

About Discount Offers and Discounts

A discount offer is a purchasable product offering, similar to a charge offer. You can use discount offers to do the following:

■ Reduce a charge by a percentage. For example, reduce a monthly fee by 50%.
■ Track usage or spending by using counters, and discount the charge for an event based on the value of the counter. For example, a discount can track usage and apply a 25% discount to the charge of an event when over 1000 minutes have been used.
About Discount Offers and Discounts

- Consume or grant balance impacts. For example, if an event is charged one dollar per minute, you can use a discount offer to allow the customer to use their included minutes and credit the corresponding charge. The discount offer would include a discount with two balance impacts:
  - Consume one minute for every dollar charged.
  - Credit one dollar for every minute consumed.

A discount offer includes one or more discounts. Every discount has one or more balance impacts. The balance is represented by a currency or noncurrency balance element. A debit balance must be a noncurrency balance element, such as minutes.

Figure 8–1 shows a balance impact for a 10 percent discount on a monthly recurring charge.

**Figure 8–1   Discount Reducing a Charge by a Percentage**

<table>
<thead>
<tr>
<th>Impact</th>
<th>Balance Element</th>
<th>Amount</th>
<th>Per Unit</th>
<th>What to Discount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>US Dollar</td>
<td>10.00</td>
<td>Percent of</td>
<td>Charge</td>
</tr>
</tbody>
</table>

A discount is calculated based on two elements in the balance impact, the values in the *Per Unit* column and the *What to Discount* column. The *Per Unit* value is how to apply the balance element amount to the *What to Discount* value.

In Figure 8–1, the *Per Unit* value is a percentage, applied to the charge (the *What to Discount* value). Because the event is charged in US Dollars, the balance impact is also in US Dollars. A currency balance element must be the same balance element in the charge that the discount is granted to.

The *Per Unit* value can be a percentage, an absolute value, or incremental; for example, apply the discount for every 10 gigabytes downloaded. The *What to Discount* value can be based on a charge, a quantity, a balance, or a calculation of multiple values.

Figure 8–2 shows a discount that grants 5 minutes for every 10 gigabytes of data downloaded. The *What to Discount* value in this case is the value that is used to calculate the discount by applying the *Per Unit* value.

**Figure 8–2   Grant 5 Minutes for Every 10 Gigabytes**

<table>
<thead>
<tr>
<th>Impact</th>
<th>Balance Element</th>
<th>Amount</th>
<th>Per Unit</th>
<th>What to Discount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>Included Minutes</td>
<td>5.00</td>
<td>For every 10 of</td>
<td>Balance [Gigabytes Counter]</td>
</tr>
</tbody>
</table>

Figure 8–3 shows a discount that credits $5 on a purchase fee. In this case, the *Per Unit* value is Fixed, so the *What to Discount* value is not needed for calculating the discount, and is set to Not Applicable.

**Figure 8–3   $5 Discount on a Purchase Fee**

<table>
<thead>
<tr>
<th>Impact</th>
<th>Balance Element</th>
<th>Amount</th>
<th>Per Unit</th>
<th>What to Discount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>US Dollar</td>
<td>5.00</td>
<td>Fixed</td>
<td>Not Applicable</td>
</tr>
</tbody>
</table>
A discount offer applies to a single service. You can also configure a discount offer to apply to the account. In that case, a discount would apply to charges for the same event for any service owned by the account. Discount offers that apply to a service are not valid when the service is inactive or canceled. Account discount offers are valid even if no service is active.

Within each discount offer, each discount has a discount type based on the events configured for the service that the discount offer applies to. Discount types are organized in discount categories as follows:

- **Usage**: These discount types discount usage events. For example, you might use the following:
  - The ConvergentUsage discount type to discount Wireless Voice events.
  - The ConvergentData discount type to discount Wireless Data events.

- **Recurring**: These discount types discount recurring events, such as a monthly fee. For example, recurring discount types might be Monthly Recurring Event and Annual Recurring Event.

- **One-time**: These discount types discount one-time events, such as purchase events or cancel events.

- **Billing Time**: This discount type is calculated when billing is run, instead of when the charge is rated. See "About Billing-Time Discounts" for information.

For recurring discounts, note the following:

- Although a charge offer cannot contain both cycle forward and cycle arrears charges, a discount offer can contain both types of cycle discounts as long as their frequency (monthly, yearly, and so on) is the same.

- A recurring discount is applied to all the recurring charges for the same service. For example, if there are different Monthly Recurring Event charges for the same service, the Monthly Recurring Event discount applies to all of them. You can use discount filters to apply different discounts to them, by selecting different pricing names. See "About Discount Filters".

When you create a discount offer in PDC, you configure settings that apply to the offer, for example, when the discount offer is available for purchase. See "Configuring How Discount Offers are Purchased and Applied".

When you create a discount, you can define the following:

- You can calculate discounts based on quantity, for example, the quantity of minutes use. See "Applying Discounts Based on Quantity".

- You can use filters to specify criteria that a charge must satisfy to be eligible for the discount. See "About Discount Filters".

- You can use discount triggers to restrict the application of a discount until certain conditions are met. See "About Discount Triggers".

- You can use the results of one discount to calculate another discount for the same event. See "About Using Temporary Balances in Discounts".

- You can calculate discounts based on data in the BRM database, or other data that is not directly configurable in PDC. See "Using Functions to Calculate a Balance Impact".

- You can use rules to apply multiple filters, triggers, and pricing to the same event. See "Applying Multiple Discount Versions to an Event".
You can use counters to create a value that can provide a discount when billing is run. See "About Billing-Time Discounts."

You can use date ranges to apply different discounts based on a fixed date range. See "Configuring Effective Dates for Discounts."

You can configure how a discount is applied when it is valid for only part of a cycle. See "Configuring Discount Validity."

You can create a billing-time discount that distributes a percentage discount to all accounts in a discount sharing group. See "About Snowball Discounts."

Configuring How Discount Offers are Purchased and Applied

You can control how a discount offer is purchased and applied by configuring the following:

- You can create subscription discount offers and system discount offers:
  - **Subscription:** The discount offer applies only to the customer who owns the discount offer.
  - **System:** The discount offer applies to all customers who use the specified service. Use system discounts when you do not want to re-create the same discount multiple times in your product offerings.

- You can configure a range of dates during which the discount offer can be purchased. To be added to a bundle, a discount offer must have a purchase period that is the same as or greater than the bundle’s purchase period.

  You can also specify the effective period for the discount offer when you configure the bundle that the discount offer belongs to. For example, you can set the discount to be effective one month after it is purchased. See "About the Validity Periods of Offers in Bundles".

- You can specify how many instances of the discount offer that the customer can purchase and own.

- You can assign a priority number to each discount offer. Offers with higher priority numbers are applied before those with lower priority numbers.

- You can specify how to apply multiple discount offers to a single charge. See "Applying Multiple Discount Offers to an Event" for information.

- You can specify how to configure mutually-exclusive discount offers. See "Configuring Mutually Exclusive Discount Offers" for information.

Applying Multiple Discount Offers to an Event

The same charge or portion of a charge can be eligible for multiple discount offers. When this occurs, BRM uses the discount offer priority and the following options to determine how to apply the discount offers to the charge:

- **Original Charge:** The discount offer is applied to the original charge amount, regardless of whether that amount was reduced by previous discount offers.

- **Remaining Charge:** The discount offer is applied to whatever charge amount remains after previous discount offers are applied.

- **Remaining Charge and Quantity:** The discount offer is applied only to the part of the charge and quantity that were not discounted already by a previous discount offer. For this option, the discount offer can be used only if:
Configuring How Discount Offers are Purchased and Applied

Creating Discount Offers

– Part of the charge has not yet been evaluated for a discount.
– The discount offer consumes noncurrency balances or reduces currency charges.
– The discount does no grant balances.

For example, a bundle includes the following discount offers:

- First priority: 10% off
- Second priority: 20% off

The customer makes a 100-minute call at $0.10 a minute for a total charge of $10. The discount for the charge is calculated as follows:

- Using Original Charge:
  - First discount: 10% of $10 = $1 discount
  - Second discount: 20% of $10 (the original charge) = $2 discount
  - Total discount = $3
- Using Remaining Charge:
  - First discount: 10% of $10 = $1
  - Second discount: 20% of $9 (the remaining charge) = $1.80
  - Total discount = $2.80
- Using Remaining Charge and Quantity:
  - First discount: 10% of $10 = $1
  - Second discount: Not applied because there is no remaining charge that has not been discounted.
  - Total discount = $1

Configuring Mutually Exclusive Discount Offers

You configure discount offer exclusions to prevent a discount offer from being used or purchased when another discount offer or package is owned. You can use two methods to configure mutually exclusive discounts:

- Discount offer-to-discount offer exclusions. You can configure a discount offer to exclude the application of other discount offers, when both discount offers apply to the same event. You configure discount offer-to-discount offer exclusions when you create discount offers.

- Discount offer-to-package exclusions. You can configure a package to exclude specified discount offers. You specify discount-to-package exclusions when you create packages.

Discount offer-to-package exclusions can be configured to operate in two ways:

- At purchase. If an exclusion relationship exists between a discount offer and a package, a customer can own the discount offer or the package, but not both. Further, the customer cannot own any discount offers associated with the package if the customer owns the excluded discount offer.

- At run time during charging. In this case, the customer is allowed to purchase the excluded discount, but it is not applied during charging or billing.

When configuring discount exclusions, note the following:
To exclude a billing-time discount, you must also exclude the counter discount associated with the billing-time discount (see “About Billing-Time Discounts”).

Discount exclusion takes precedence over discount priority; a discount can be excluded even if it has a higher priority than the discount that is excluding it.

System discounts can be excluded.

Shared discounts can be excluded.

Discount exclusions are disabled by default. To enable discount exclusions, see the chapter on managing purchased offerings in BRM Managing Customers.

Applying Discounts Based on Quantity

To apply discounts based on usage amount or on an amount in a balance, you can create quantity ranges. For example, a discount for a mobile phone service might contain the following quantity ranges:

- 0 through 500 minutes: No discount
- 500 through 1000 minutes: 10% off
- 1000 minutes through No Maximum: 15% off

In this example, the quantity is based on how many minutes are used. You can also based the quantity on the charge, an amount in a balance, or calculated value.

With one exception, the start and end values of all quantity ranges must be a decimal. The exception is that a calculated value can be used to specify the end value of the last quantity range. For example, to use all the minutes in a balance, the calculation would be:

0 - Balance[Included Minutes]

In addition to specifying what to base the quantity on (usage, charges, and so on), you specify how to apply the discount to the quantity ranges. There are two options:

- **Pick the quantity range containing the value:** Applies the discount in the range that contains the entire quantity amount. Using the previous example, if the value is 700, the ranges are applied as follows:
  - 0–500 range does not qualify.
  - 500–1000. The discount for this range is applied to the entire result.
  - 1000–No Maximum does not qualify.

- **Distribute value across applicable quantity ranges:** The pricing in each range is applied to the amount of the basis that falls within it. If the value is 700, the ranges are applied as follows:
  - 0–500. The discount for this range is applied to the first 500 minutes.
  - 500–1000. The discount for this range is applied to minutes from 500 to 700.
  - 1000–No Maximum does not qualify.

When you distribute the value across ranges, you can use StepCharge or StepQuantity for the What to Discount value. This ensures that the discount is applied to only part of the quantity range:

- StepCharge bases the discount on the charge that applies to the quantity range, not the entire charge.
About Discount Filters

A discount filter specifies criteria that a charge or a portion of a charge must satisfy to be eligible for the discount. For example, a discount might apply only to calls made on weekends or during December.

Filter criteria can be defined by specific values (such as a time period from a specified time model) or by using regular expressions.

You can create filters based on several values; for example, the amount consumed by the event, a balance element, a time period (peak and off-peak) and so on.

Figure 8–4 shows a discount filter that applies a discount to the event if it is measured in volume (not duration) and the balance element is a prepaid balance.

When creating a discount filter:

- You can specify a date range that defines when the filter is valid.
- If a discount has multiple filters, all filters are applied.
- A filter can be reused by multiple discounts.
- You can enter NOT operator in the following fields to specify the values to be excluded:
  - GLID
  - Time Period
  - Impact Category
  - Generic Selector Result

Note: PDC does not validate the expressions entered in the fields.
About Discount Triggers

You can use discount triggers to restrict the application of a discount until certain conditions are met. For example:

- A discount granted only if the number of domestic call minutes is less than or equal to 100.
- The bytes used in a GPRS session must be greater than 1 MB.

Figure 8–5 shows a discount trigger in PDC.

![Discount Trigger](figure8-5.png)

If a trigger condition includes multiple conditions, all conditions must be met for the discount to be granted.

A discount can have only one trigger. A trigger can be used by multiple discounts.

About Using Temporary Balances in Discounts

Typically, you use temporary balances when you need the results of one discount to calculate another discount for the same event.

For example, consider a discount that awards 5 text messages if the amount of data transferred in a data usage event exceeds 1 megabyte and the session duration exceeds 30 minutes. In this situation, Discount 1 would determine the amount of data sent or received and store it in Temporary Balance A. Discount 2 would determine the duration of the session and store it in Temporary Balance B. Discount 3 would grant 5 text messages if the data in Temporary Balance A exceeds 1 megabyte and the session length in Temporary Balance B exceeds 30 minutes.

You can only use noncurrency balances for temporary balances. Rounding rules do not apply to them.

Unlike other balances, temporary balances are maintained only while a single event is being discounted. If multiple discounts are applied to a single event, the temporary balance is maintained until all the discounts are processed.

For billing-time discounts, temporary balances are maintained only while a single discount is processed. Therefore, you cannot use temporary balances for billing-time discounts to pass a balance impact from one discount offer to another.

Using Functions to Calculate a Balance Impact

Use a function in a discount to calculate a value that is not directly configurable in PDC. The function typically performs a database query, for example, the total charges for the ten most frequently called numbers. This amount can then be discounted.
For example, suppose you want to implement a 10% discount on data usage from an access point. To implement this discount, you could include a function as the What to Discount value:

**Function["AccessPoint"]**

When BRM processes this discount, the usage amounts for the access point is substituted into the What to Discount value. A 10% discount is applied to AccessPointX usage.

### Applying Multiple Discount Versions to an Event

You can apply multiple versions of a discount to the same event. To do so, you create multiple discount rules. For example, you might use discount rules to apply a discount in peak or off-peak times. A discount rule includes all of the elements of a discount; filters, triggers, and pricing. The rules are applied in the order that they appear in PDC.

You can use the same criteria (Original Charge, Remaining Charge, and Remaining Charge and Quantity) for applying multiple discount rules as you do when you apply different discount offers to the same event. See "Applying Multiple Discount Offers to an Event."

### About Billing-Time Discounts

Billing-time discounts are calculated at the end of the billing cycle. This enables you to apply discounts based on the aggregation of a balance during a billing cycle. For example, you can create a billing-time discount to do the following:

- Reduce a balance by $10 if the total usage charges for the billing cycle are more than $100.
- Grant 10 text messages if the total minutes used during the billing cycle are more than 500.

Billing-time discounts are typically based on balances accumulated by using a counter. The counter can be used to apply usage, recurring, or one-time discounts.

For example, you can create a billing-time discount that grants 10 minutes for every $100 in usage charges. To create a billing-time discount, you set up the following discounts:

- A non-billing-time discount that increments a counter to track Dollars Spent. Every $1 charged increments the counter by 1.
- A billing-time discount grants 10 minutes if the Dollars Spent counter is over 100 at billing time.

### Configuring Effective Dates for Discounts

You can add a date range to a discount by specifying a fixed start date and end date (discount date ranges cannot be relative). The date range is the period during which the discount is effective.

This enables you to use different versions of the same discount during different time periods by adding multiple date ranges to a single discount rather than by creating multiple discounts.
You can use different discount rules and pricing for each date range. For example, you could create a discount that applies a 10% discount for the first date range and a 20% discount for the second date range.

Date ranges cannot overlap in the same discount. For example:

- Immediately through 6/1/2016
- 6/1/2016 through 1/1/2017
- 1/1/2017 through never ends

---

**Note:** Date ranges for different discounts in the same discount offer can overlap.
Configuring Discount Validity

You can configure discount validity for the following cases:

- **Applying Discounts That Are Valid for Only Part of a Cycle**
- **Applying a Discount After the Discount Offer Is Inactivated or Canceled**
- **Configuring Validity Periods for Noncurrency Credit Balance Impacts**

### Applying Discounts That Are Valid for Only Part of a Cycle

You can specify how to apply a discount when it is valid for only part of a cycle. There are three scenarios:

- Discounts that are valid from mid-cycle to the end of the cycle. *Mid-cycle* is any time after the accounting cycle start time and before the accounting cycle end time.
- Discounts that are valid from the start of a cycle but end in mid-cycle.
- Discounts that are valid for part of a cycle, in the middle of the cycle.

For each scenario, you specify whether the discount is applied for the full cycle, applied to part of the cycle, or is not applied at all.

For example, Figure 8–6 shows a discount that becomes valid in mid-cycle, but is applied to the entire cycle.

---

**Note:** In this example, if the discount is a usage discount, the charges would need to be rerated to apply the discount to the events that occurred before the discount was valid. See the discussion of rerating in *BRM Implementing Charging*.

---

**Figure 8–6  Valid Mid-cycle - Full Discount**

<table>
<thead>
<tr>
<th>Cycle start</th>
<th>Discount activated</th>
<th>Cycle end</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1</td>
<td>1/1/5</td>
<td>2/1</td>
</tr>
</tbody>
</table>

**Figure 8–7** shows a discount that becomes valid in mid-cycle that is applied to only part of the cycle.

**Figure 8–7  Valid Mid-cycle - Partial Discount**

<table>
<thead>
<tr>
<th>Cycle start</th>
<th>Discount activated</th>
<th>Cycle end</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1</td>
<td>1/1/5</td>
<td>2/1</td>
</tr>
</tbody>
</table>

**Figure 8–8** shows a discount that becomes valid in mid-cycle, but no discount is applied.
**Figure 8–8  Valid Mid-cycle - No Discount**

<table>
<thead>
<tr>
<th>Cycle start</th>
<th>Discount activated</th>
<th>Cycle end</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/1</td>
<td>1/5</td>
<td>2/1</td>
</tr>
</tbody>
</table>

No discount applied for the cycle. Discount starts on 2/1

**Applying a Discount After the Discount Offer Is Inactivated or Canceled**

For each discount, you specify whether to apply the discount after the discount offer is canceled or inactivated. This controls how to apply discounts to events that occurred prior to the discount offer cancellation or inactivation. The options are:

- Don’t stop discounting if the discount offer is inactive or canceled
- Stop discounting if the discount offer is inactivated
- Stop discounting if the discount offer is canceled
- Stop discounting if the discount offer is inactivated or canceled

**Configuring Validity Periods for Noncurrency Credit Balance Impacts**

When you create a discount balance impact that credits noncurrency balances, such as granting minutes, you can configure a validity period for the balance impact. This is useful if you want to limit how long a customer can keep minutes in their balance before they need to purchase more minutes. Figure 8–9 shows a balance impact that grants 100 minutes that are available for one billing cycle.

**Figure 8–9  Balance Impact for Minutes**

<table>
<thead>
<tr>
<th>Impact</th>
<th>Amount</th>
<th>Balance Element</th>
<th>Amount Is Valid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credit</td>
<td>100.00</td>
<td>Included Minutes</td>
<td>From first usage - 1 Billing Cycles From Validity Start</td>
</tr>
</tbody>
</table>

You can specify a validity period corresponding to the current accounting cycle or the current billing cycle. You can also define validity period start and end times. You can start a validity period:

- When the event occurs.
- At first usage. Balances that start on first usage are added to the account balance at the time of the grant, but the validity starts when the subscriber uses the service for the first time.
- From a specified date.
- Relative to a specified date.

You can end a validity period based on:

- A date relative to the start date (measure in minutes, hours, days, months, or in account, billing, or recurring cycles)
- On a specified date.
- No end date.
If a charge offer or discount offer is canceled after being used, the validity end time of any balance granted by the offer is set to the time of the cancellation.

**Discount Offers and Tax Calculation**

You can apply a tax code to a discount. You need to define tax codes in BRM before you can assign them in discounts.

For information, see *BRM Calculating Taxes*.

**Discount Offers and General Ledger**

You assign a general ledger ID (G/L ID) to discount balance impacts. This enables you to track revenue for each type of discount.

**About Snowball Discounts**

A snowball discount is a type of shared billing-time discount that distributes a percentage discount to all accounts in a discount sharing group. A discount sharing group allows one account (the group owner) to share discounts with other accounts (the group members).

To implement a snowball discount, you use two discounts:

- A discount that increments a counter.
- A billing-time discount that calculates and distributes the discount based on the counter balance. In PDC, you designate this a snowball discount.

For example, a discount sharing group has one owner and three members. The owner purchases a discount offer that includes a snowball discount that grants $0.01 for every minute of telephone usage. At the end of the billing cycle, the group has used a total of 4,000 minutes. A total of $40 is granted to the group. If the discount is configured to divide the grant evenly, the owner and each member receives $10.

You can also configure the discount to distribute the grant based on the number of minutes used for each account. For example, if the owner uses 2,500 minutes, and each member uses 500 minutes, the owner receives a $25 grant, and each member receives a $5 grant.

In addition to creating the counter discount and the billing-time discount, you need to do the following:

- Set up a discount sharing group by defining the group owner, group members, and a list of the discounts that will be shared by the group. See "About Discount Sharing" for more information.
- Specify how the snowball discount is distributed in the pin\_snowball\_distribution file and load the file by running the load\_pin\_snowball\_distribution utility.

  The distribution you specify in the pin\_snowball\_distribution file becomes the default distribution. You can customize this behavior by modifying the policy opcode PCM_OP_SUBSCRIPTION\_POL\_SNOWBALL\_DISCOUNT.

- If the discount is to be distributed unevenly, for example, if each member gets a percentage of the discount based on its own usage, configure sub-balance contributors. Specify the contributors in the pin\_sub\_bal\_contributor file and load the file by running the load\_pin\_sub\_bal\_contributor utility. See "Defining and Loading Custom Sub-Balances" for information.
Defining How Snowball Discounts Are Distributed

To specify how snowball discounts are distributed, edit the `pin_snowball_distribution` file, and then run the `load_pin_snowball_distribution` utility to load the contents of the file into the `/config/snowball_distribution` object in the BRM database.

You set two values in the `pin_snowball_distribution` file:

- The name of the discount offer.
- Either zero or the balance element ID for the counter discount.
  - If set to zero, the discount is distributed evenly among all members of the discount sharing group.
  - If set to a balance element ID, the discount is distributed based on how much each account contributed to the amount in the counter balance.

**Note:** If you specify a balance element, you must configure sub-balance contributors to enable BRM to track each account’s contribution. See “Defining and Loading Custom Sub-Balances” for information.

For example, the following entries in the `pin_snowball_distribution` file specify that `snowball one` is distributed evenly and `snowball two` is distributed based on the amount each account contributed to the counter (balance element ID 1000501) at the end of the billing cycle.

```
snowball one : 0
snowball two : 1000501
```

**Caution:** The `load_pin_snowball_distribution` utility overwrites existing snowball distribution rules. If you are updating snowball distribution rules, you cannot load new snowball distribution rules only. You must load complete sets of snowball distribution rules each time you run the `load_pin_snowball_distribution` utility.

To configure snowball distribution rules:

1. Edit the `pin_snowball_distribution` file in `BRM_home/sys/data/pricing/example`. The `pin_snowball_distribution` file includes instructions.
2. Save the `pin_snowball_distribution` file.
3. Use the following command to run the `load_pin_snowball_distribution` utility:

   ```shell
   load_pin_snowball_distribution pin_snowball_distribution
   ```

   If you do not run the utility from the directory in which the file is located, you must include the complete path to the file, for example:

   ```shell
   load_pin_snowball_distribution BRM_home/sys/data/pricing/example
   ```

**Tip:** If you copy the `pin_snowball_distribution` file to the directory from which you run the `load_pin_snowball_distribution` utility, you do not have to specify the path or file name. The file must be named `pin_snowball_distribution`.
4. Stop and restart the Connection Manager (CM).

To verify that the network elements were loaded, you can display the 
/config/snowball_distribution object by using the Object Browser, or use the robj 
command with the testnap utility. See "Reading an Object and Writing Its Contents to 
a File" in BRM Developer’s Guide.
This document describes how to create Oracle Communications Billing and Revenue Management (BRM) bundles in Pricing Design Center (PDC).

Topics in this document:

- About Creating Bundles
- About the Validity Periods of Offers in Bundles
- About Bundles and First-Usage Balance Impacts
- Specifying Discounts in Bundles
- Creating Dependencies for Bundles
- Transitioning between Bundles

See also:

- About Creating Product Offerings
- Configuring Charge Offers
- Creating Discount Offers
- Creating Packages and Package Lists

**About Creating Bundles**

A bundle is a set of charge offers, discount offers, or both. Bundles are typically used to group offers that you want to sell together.

Each bundle is associated with a single service. Only offers that apply to that service can be included in the bundle. For example:

- A package that provides only broadband access includes only broadband access bundles (such as cable and premium cable).

- A package that provides broadband access and VOIP includes at least two bundles, one for broadband access and one for VOIP.

- A package that provides VOIP and cable TV includes at least two bundles, one for VOIP and one for cable TV.

*Figure 9–1* shows bundles that are associated with a single service.
One bundle can contain any number of charge and discount offers, and two different bundles can contain the same offer. Grouping offers in different ways in different bundles adds flexibility to your pricing structure without requiring you to create additional charge or discount offers.

You have the following options when creating bundles:

- **Associate the bundle to a service or to Account.** When creating a bundle, you can associate it with a specific service. That is the only service to which the offers in the bundle apply.

  You can also associate the bundle with Account. For example, you might do this to create a bundle for late charges or for coupons. The offers in the bundle can then be used to rate any event associated with Account in the service-event map. Each account can own only one bundle that applies to Account.

- **Specify when a bundle can be purchased.** By default, the time period during which a bundle can be purchased starts immediately and never ends. You can change the default start date, end date, or both.

  To be added to a bundle, an offer must have a purchase period that is the same as or greater than the bundle purchase period. If the purchase period of an offer in a bundle exceeds the bundle purchase period, the bundle purchase period overrides the offer purchase period.

- **Bill customers when they purchase the bundle.** Usually, you bill a customer at the end of the customer’s billing cycle. On-purchase billing, however, enables you to bill a customer immediately for a purchase, even if the customer’s billing cycle has not ended. When you create a bundle, you can flag it for on-purchase billing. When a customer purchases the bundle, a bill is generated immediately for the purchase fees associated with the bundle.

  **Note:** On-purchase billing works with purchase fees only, not with recurring, usage, or cancellation fees.

- **Permit CSRs to customize bundles.** If your CSRs (customer service representatives) can discount or change the effective period of charges in bundles at purchase time, you can specify whether to prohibit, allow, or require such modifications in a particular bundle. For example, if customer input is required to
set the date on which an offer’s purchase fee is applied, you can specify that a bundle must be modified.

- **Configure offer quantities.** You can configure bundles to provide more than one of the same charge offer or discount offer. For example, if a bundle for a cable service includes a charge offer that provides one set-top box and you want to include three set-top boxes with the cable service, enter 3 for the charge offer in that bundle.

- Configure the status of offers at purchase. When you add a charge offer or discount offer to a bundle, you also specify whether the offer is active or inactive at the time of purchase. For example, an offer might be inactive at purchase so that the purchase or first month's fee is not applied until you get confirmation that the hardware was received and successfully configured. By default, offers are active.

  For offers with inactive status, you can specify a reason for that status.

See the following topics for more bundle configuration features:

- About the Validity Periods of Offers in Bundles
- About Bundles and First-Usage Balance Impacts
- Specifying Discounts in Bundles
- Creating Dependencies for Bundles
- Transitioning between Bundles

### About the Validity Periods of Offers in Bundles

You can configure the start time and end time for each charge offer and discount offer in a bundle. The time between the start time and the end time is called the **validity period**. A customer cannot use a service provided by an offer unless the offer is valid.

**Notes:**

- You specify when a charge offer or a discount offer can be purchased in the offer itself. You cannot apply a validity period when you create a charge offer or a discount offer; you can only apply it in the bundle configuration.

- You can apply validity periods to individual charges and discounts configured within offers. However, the validity periods specified in a bundle take precedence over the validity periods specified in the charges and discounts. For example, you might want to allow a customer to purchase a charge offer, but not allow the customer to use the service until it is activated on the network.

In bundles, you set the validity period of offers by specifying the start and end times of the offer and of the offer’s cycle and usage charge periods.

- **Offer start and end times:** Specify when a customer can use the service or benefit from the discount. The offer start time is when the purchase fee is charged. It is also the earliest time that the offer’s fees can begin to accumulate in an account balance.

- **Recurring and usage charge periods:** Specify when recurring and usage events can be charged or discounted. These validity periods must not begin before the offer start time.

You can set *start* times as follows:
About Bundles and First-Usage Balance Impacts

- **Immediately:** (Default) The offer or charge is valid and can be activated immediately. The purchase fee is charged as soon as the offer is added to the account.

- **Relative to Purchase:** The purchase fee is charged when the offer is added to the account, but the customer cannot use the service or benefit from the discount until the relative period ends.

When a charge offer’s charge period has a relative start time, the events are not rated and the fees are not charged until the relative period ends, even if the service has been activated. This option enables you to waive subscription or usage fees for a period of time.

When a discount offer’s charge period has a relative start time, the discount is not applied to cycle or usage fees until the relative period ends.

- **First Usage:** The offer is activated and the purchase fee is charged when the customer first uses the service, such as by making a phone call. The charge periods also begin at that time. The validity period start time is set to the start time of the event that first uses the service or triggers the discount offer. The end time is set based on the end time that you configure for the charge offer or discount offer.

Setting charge offers and discount offers to start when they are first used enables you to delay charging customers for the services they purchase until they start using those services or to delay activating discount offers until they can be applied to customers’ usage. This is useful when you offer limited-time services or discount offers that expire relative to when they are activated.

You can also set up individual noncurrency balances granted by charge offers and discount offers to start when the balances are first consumed. However, there is no dependency between a discount offer’s first usage start time and the first usage start time of a balance it grants. For example, a discount offer might start based on a different balance impact than the balance impact flagged as starting on first usage.

**Note:** Charge offers that start on first usage must include usage fees. If you set a charge offer that has no usage fee to start on first usage, the charge offer will never be activated.

You can set the *end* times as follows:

- **Never:** (Default) After it is activated, the offer is effective indefinitely, and its recurring and usage fees can be charged or discounted indefinitely.

- **Relative to Start:** After the relative period ends, the offer is not effective and the recurring or usage fees are not charged or discounted.

If the offer end time specified in the bundle is earlier than the recurring or usage charge end time, the offer end time overrides the charge end time.

**About Bundles and First-Usage Balance Impacts**

See these topics:

- About Activating First-Usage Discount offers
- About Granting First-Usage Balance Impacts In First-Usage Charges
- Aligning Validity of Balance Impacts on First Usage
About Activating First-Usage Discount offers

Discount offers that start on first usage can be activated when the customer first uses a service, when the first cycle fee is applied, or when the account’s bill is generated, depending on which fees are discounted and when the discount is triggered. For example, a first-usage discount on SMS messaging is activated when the customer sends the first SMS message, a first-usage discount on cycle fees is activated when the first cycle fee is applied, and a first-usage billing-time discount is activated when the first bill is generated for the account.

It is possible that a discount on a particular service might not be activated when a customer first uses that service. For example, if long-distance calls are discountable for a telephony service, a customer might make multiple local calls, which do not trigger the discount’s effective period, before making a long-distance call.

There are some cases in which a discount’s balance impact is negated by a second discount. If this occurs, the first discount’s validity period is still set.

For example, a customer purchases a package that includes discount offer A and discount offer B:

- Discount Offer A is configured to start when first used, gives 10% off of all calls, and has a higher priority, so it is applied first.
- Discount Offer B is configured to start immediately and makes all birthday calls free.

If a customer makes a first-usage call on his birthday and is charged $5.00 for the call, discount offer A is applied first, which reduces the balance by $.50, and discount offer A’s validity period is set. Then discount offer B backs out all charges. In this case, the validity period of discount offer A remains set.

About Granting First-Usage Balance Impacts In First-Usage Charges

A charge offer that is valid from first usage should not grant a credit balance, such as included minutes, that is also valid from first usage. If it does, the granted balance cannot be consumed by the first usage event.

For example, consider a charge offer that is valid on first usage and has these charges:

- A recurring event that grants minutes. The balance impact is set to first usage.
- A charge the consumes minutes until there are none, and then consumes money.

When the first usage event for this charge offer occurs, the charges are processed in this order:

1. The first usage event is charged, and it consumes a monetary balance, because no minutes have been granted.
2. The recurring event, with a balance impact set to first usage, can now be charged. It grants minutes.
3. Subsequent usage events now have minutes to consume.

Because the included minutes are valid from first usage, they are not granted until after the first usage event is rated. Hence, the first usage event cannot consume any of the minutes included in the charge offer. Therefore, the customer is charged for a call that should have consumed minutes.

However, if a discount offer that is valid from first usage grants a credit balance that is valid at first usage, the granted balance is available for consumption because discounting occurs after the balances are granted.
Aligning Validity of Balance Impacts on First Usage

When you create a bundle, you can flag it to synchronize the start date of all balance impacts whose validity period starts on first usage. This ensures that all such balance impacts in the bundle’s charge offers are set to the same validity period when one of them is activated for the first time.

Specifying Discounts in Bundles

In addition to adding discount offers to a bundle, you can reduce one-time and recurring charges in charge offers by specifying a flat percentage discount for each charge category in the bundle itself. For example, you can discount all recurring charges and one-time charges by 10%. You cannot use a bundle to apply discounts to usage charges.

When you discount a charge category in a bundle, the discount applies to all charge types that apply to that charge category. For example, you might have a charge offer that includes two recurring charges, Monthly Fee and Grant Minutes. If you specify a percentage discount for recurring charges in the bundle, the discount applies to both recurring charges. To discount only the Monthly Fee charge, you must make the balance impact for Monthly Fee discountable and the balance impact for Grant Minutes nondiscountable.

Unlike discount offers, discounts configured in bundles cannot be tracked in the general ledger because they are not associated with a general ledger (G/L) ID. In addition, unlike discount offers, bundle-configured discounts are difficult to display on a customer’s bill because they are not separate items.

---

**Note:** Because of the limitations associated with discounts configured in bundles, Oracle recommends that you use discount offers instead. See "Creating Discount Offers" for more information.

Creating Dependencies for Bundles

You can set up the following dependent relationships between bundles:

- **Prerequisite:** Specifies that an account must own a particular bundle to be able to purchase another particular bundle. A prerequisite can include bundles for different services. For example, to own a GPRS bundle, an account might be required to own a GSM bundle.

- **Mutually Exclusive:** Sets up a mutually exclusive relationship between two bundles so that if an account owns one of the bundles, it cannot own the other. For example, if you set up a mutually exclusive relationship between a Corporate Voice bundle and a Residential Voice bundle, customers who purchase one cannot purchase the other.

Transitioning between Bundles

You can configure rules for transitioning from one bundle to another. PDC supports the following types of transitions:

- **Upgrade** to a bundle that is typically more expensive and has more features.

- **Downgrade** to a bundle that is typically less expensive and has fewer features.
Transition rules enable you to limit the bundles that customers can transition to and remain fully provisioned.

While transitioning from one bundle to another, your customers retain their devices, such as phone numbers, and their services.

---

**Note:** Transitioning between bundles affects only the charge offers and discount offers related to the service to which the bundle applies; the service itself remains the same. To define a transition that adds a new service, you create a transition for a package (see “Transitioning between Packages”).

---

To customize bundle transitions, use the PCM_OP_CUST_POL_TRANSITION_DEALS and PCM_OP_CUST_POL_TRANSITION_PLANS opcodes. See BRM Opcode Guide.
Creating Packages and Package Lists

This document describes how create Oracle Communications Billing and Revenue Management (BRM) packages and package lists in Pricing Design Center (PDC).

Topics in this document:
- Creating Packages
- Grouping Services
- Transitioning between Packages
- Creating Package Lists

See also:
- About Creating Product Offerings
- Configuring Charge Offers
- Creating Discount Offers
- Creating Bundles

Creating Packages

A package is a collection of bundles. You use packages to offer your services to customers. For example, if your company sells broadband access, email, and IP fax service, you might offer the following packages:

- A package that sells only broadband access
- A package that sells broadband access and email
- A package that sells email and IP fax service

Figure 10–1 shows how you can include bundles and the services to which they apply in a variety of packages.
A package can contain any number of bundles, and two different packages can share the same bundle. By grouping bundles into packages, you simplify the choices presented to customers.

You have the following options when creating packages:

- **Bill customers when the package is purchased.** Usually, you bill a customer at the end of the customer’s billing cycle. On-purchase billing, however, enables you to bill a customer immediately for a purchase, even if the customer’s billing cycle has not ended. When you create a package, you can flag it for on-purchase billing. When a customer purchases a package that is flagged for on-purchase billing, a bill is generated immediately for the purchase fees associated with the package.

  **Note:** On-purchase billing works with purchase fees only. It does not work with recurring, usage, or cancellation fees.

- **Configure required and optional bundles in a package.** **Required bundles** are automatically purchased when a package is purchased. By default, all bundles are required. **Optional bundles** can be purchased at a later time.

- **Restrict discounts in a package.** When creating a package, you can prohibit specified discount offers from being purchased or used while the package is owned.

  If an exclusion relationship exists between a discount offer and a package, a customer can own the discount offer or the package, but not both. Further, the customer cannot own any discount offers associated with the package if the customer owns the excluded discount offer.

  See “Configuring Mutually Exclusive Discount Offers” for information about configuring discount exclusion.

See the following topics for more package configuration features:

- **Grouping Services**
- **Transitioning between Packages**
Grouping Services

In packages, you can combine services into service groups. A service group consists of one subscription service, and one or more member services. For example, if the subscription service is GSM, the member services could be Voice and SMS.

Service groups provide the following benefits:
- Member services can benefit from discounts owned by the subscription service.
- Member services can be associated with the devices owned by the subscription service.
- Customers can purchase services as a group.

The subscription service can be a service that subscribers use, such as telephony or broadband access, or it can be a representational service with no associated usage fees. Together, the subscription service and member services form a service group.

For example, in Figure 10–1, the GSM service is a subscription service, and SMS and Voice are its member services.

Grouped services usually apply to a particular device, such as a cable box. An account can contain multiple service groups.

To subscribe to a service group, customers subscribe to its subscription service.

Transitioning between Packages

You can configure rules for transitioning from one package to another. Each transition rule applies to a particular service. The package being transitioned to and the package being transitioned from must both contain that service, though each package can contain additional services that the other package does not contain.

PDC supports the following types of transitions:
- **Upgrade** to a package that is typically more expensive and has more features. For example, a customer might transition from a package that provides broadband and cable TV services to a package that provides broadband, cable TV, and VOIP services. That transition adds a service and a bundle and possibly changes the bundles for the existing services.
- **Downgrade** to a package that is typically less expensive and has fewer features.

Transitions enable you to limit the packages that customers can switch to while remaining fully provisioned. When transitioning to a designated package, your customers retain their devices, such as phone numbers, and their services.

Defining Generation Changes

A generation change enables you to transition customers between 2G (second generation) and 3G (third generation) wireless packages and services. Packages are called 2G or 3G depending on whether the wireless service they provide runs on a 2G or 3G wireless network.

When creating a 2G or 3G package, you can set up generation change rules to specify which packages can replace the package when it is transitioned to a different generation.

For generation changes, the packages do not have to provide the same service.
Creating Package Lists

A package list is a group of packages that is usually offered to a single type of customer. For example, you might create the following package lists:

- A package list that includes packages for customers above a certain age.
- A package list that includes packages for customers in a particular location, such as Canada.
- A package list that includes promotional discounts offered for a limited time.

You can create any number of package lists, and each package list can contain any number of packages. Different package lists can contain the same package.

The package list does not have to include all your packages. You can create packages and not include them in a package list until you need them. Or you can offer one set of packages to one group of potential customers and another set of packages to another group.

To make a package available for customers to purchase, you must include the package in a package list.

You can assign the following statuses to a package list:

- **Active**: Use this status to indicate that customers can purchase packages in the list as soon as the list is added to your system.
- **Inactive**: Use this status to indicate that the list is not visible to customers and customers cannot purchase packages from the list.

Specifying the Package List Segment and Type

The package list segment identifies the customer segment that the package list is offered to, and the package list type identifies the type of packages that are added to the package list.

The package list segment and type determines the package list name. By default, the package list segment must be **CSR**. The package list type can be either **New** or **Add-on**. For example:

- The **CSR-New** package list contains packages that create accounts and add the services and bundles in the packages to the new accounts.
- The **CSR-Add-on** package list contains packages that add services and bundles to existing accounts.

The package list segment and type are case sensitive and together uniquely identify package lists. For example, **CSR-New** and **CSR-new** are two different package lists.
Sharing Charges and Discounts

This document describes how to set up charge sharing for Oracle Communications Billing and Revenue Management (BRM) in Pricing Design Center (PDC).

Topics in this document:

- About Sharing Discounts and Charges
- About First-Usage Start Time for Shared Balances
- About Discount Sharing
- About Chargeshare Offers

See also:

- About Creating Product Offerings
- Configuring Charge Offers
- Creating Discount Offers

About Sharing Discounts and Charges

BRM enables accounts to share discounts and charges by joining groups that consist of an owner account and one or more member accounts. A sharing group can be one of the following types:

- **Discount sharing group:** The owner shares its discounts with the members. Figure 11–1 shows a group owner providing discounts to three group members:

*Figure 11–1  Discount Sharing Group*

See "About Discount Sharing" for more information.
Charge sharing group: The owner assumes charges that are incurred by the members. Figure 11–2 shows a group owner receiving charges from three group members:

Figure 11–2  Charge Sharing Group

See "About Chargeshare Offers" for more information.

About First-Usage Start Time for Shared Balances

If a balance is shared among accounts in a sharing group, the balance validity period is set when any account in the group first impacts the balance. Because the same balance is shared with all accounts in the group, the validity period of that balance applies to all accounts.

About Discount Sharing

Discount sharing occurs when an account shares its discounts with other accounts. For example, a group of employees might share a pool of minutes in their company’s mobile phone account, or a parent might purchase discounts on her email service and want those discounts to apply to her children’s email services as well.

To share discounts in BRM, you create discount sharing groups. You set up the discounts in PDC.

BRM supports shared discounts by enabling a discount to apply either to the account that generates the event (the discount user), to the account that owns the discount, or to both.

With discount sharing, you sometimes need several discounts to specify how balances are granted or consumed for each account. For example, to offer 20% discount on usage to each member account when the total usage for all member accounts exceeds 1,000 minutes, you set up a discount with a counter to track the usage for each account and another discount to apply the percentage off based on the aggregated usage recorded in the counter balance.

About Chargeshare Offers

Charge sharing enables an account to sponsor the charges of other accounts. The owner account in a charge sharing group receives the balance impact of sponsored charges incurred by the member accounts. For example, charge sharing enables a company to pay for all of its employees’ mobile phone services or a parent to pay for his child’s SMS and email services.
To share charges in BRM, you create charge sharing groups. You then set up chargeshare offers and chargeshares in PDC to determine how charges are shared among the members of the charge sharing groups.

A chargeshare uses rules that contain filters to determine whether an event qualifies for charge sharing, triggers to specify conditions that must be met before the chargeshare applies, and pricing to determine the charge sharing amounts and balance impacts.

Setting up chargeshare offers and chargeshares is similar to setting up discount offers and discounts. See "Creating Discount Offers" for more information.
This part describes various charging scenarios you can configure using Oracle Communications Billing and Revenue Management (BRM) pricing components and setup components in Pricing Design Center (PDC).

Part II contains the following chapters:

- Charging Based on Date and Time
- Charging Based on Event Attributes
- Implementing Recurring Charges
- Advanced Configuration for Recurring Charges
- Working with Extended Rating Attributes
- Configuring Closed User Groups
- Configuring Policy Specifications
This document describes how to charge for usage events based on date and time in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Charging Based on Date and Time
- About Special Day Calendars
- Rating Events Split across Time Periods

See also:

- About Creating Product Offerings
- Configuring Charge Offers

About Charging Based on Date and Time

You can configure charge offer pricing to charge different amounts depending on the date and time. For example, you can create different pricing for peak and off-peak times. To do so, you use time models.

A time model is a set of time periods. Each time period contains one or more time segments. A time segment represents a particular time, such as a day of the week or a range of several hours.

By default, a time period is immediately and forever effective. To add effective periods to a time period, you specify only the start date of the new period. That date becomes the end date of the previous period. The end date of the final effective period is always Forever.

You can define time segments by using months, days of the month, days of the week, time of day, and a calendar of special days. For example, Figure 12–1 shows an Offpeak and a Peak time period whose time segments are defined by days of the week and time of day.
About Special Day Calendars

A special day calendar is a set of dates, such as holidays, for which you want to charge special prices for your services. Each date must be one of the following types:

- **Fixed**: A specific date valid only in *one year*, such as May 8, 2011, for Mother’s Day in the U.S.
- **Recurring**: A date that is valid *every year*, such as July 4 for Independence Day in the U.S.

*Figure 12–2* shows a special day calendar that has fixed and recurring dates.

To configure pricing for special days in a charge, you associate a special day calendar with a time model. You must then configure at least one time period that applies to the special days. The time model should cover all 24 hours of the special days.
Rating Events Split across Time Periods

Events sometimes overlap time periods. For example, if off-peak rating starts at 7:30 p.m., and a call begins at 7:10 p.m. and ends at 7:35 p.m., the call overlaps the boundary between peak and off-peak rates.

When an event crosses multiple time periods, you can specify the way to charge for an event if it is priced differently by time period:

- **Start Time**: Applies the charges defined in the time period in which the event begins.
- **End Time**: Applies the charges defined in the time period in which the event ends.
- **Split**: Applies charges according to the time period in which each portion of the event falls.

If you split charges across time periods, and you configure pricing based on quantity, you can specify the way a price step is selected when part of the event was already charged in another time period.

- **Dependent On**: Dependent on the quantity already used. When selecting a price step, BRM considers the amount of the event that occurred in the previous time period.
- **Independent Of**: Independent of the quantity already used. When selecting a price step, BRM does not consider the amount of the event that occurred in the previous time period.

For example, you might create pricing such as:

- Off-peak pricing between 8 PM and 6 AM.
- During peak time, charge 25 cents per minute.
- During off-peak time, charge:
  - $0.10 per minute for the first hour
  - $0.05 per minute for subsequent hours

A phone call starts at 7 p.m. and ends at 10 p.m. This results in one hour of peak time, and two hours of off-peak time.

- If the selection quantity is **Dependent On** the quantity already used, the hour charged as peak time is considered as the first hour, and the entire off-peak portion of the call is charged $0.05 per minute.
- If the selection quantity is **Independent Of** the quantity already used, the hour charged as peak time is **not** considered as the first hour, and the off-peak portion of the call is charged $0.10 per minute for the first hour, and $0.05 for the second hour.
Figure 12–3 shows how the call is charged.

**Figure 12–3**  *Quantity-Based Pricing Split by Time Periods*

<table>
<thead>
<tr>
<th>Dependent On</th>
<th>Price per minute</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 PM</td>
<td>0.05</td>
</tr>
<tr>
<td>7 PM</td>
<td>0.10</td>
</tr>
<tr>
<td>8 PM</td>
<td>0.25</td>
</tr>
<tr>
<td>9 PM</td>
<td></td>
</tr>
<tr>
<td>10 PM</td>
<td></td>
</tr>
</tbody>
</table>

Peak  Off-peak

<table>
<thead>
<tr>
<th>Independent Of</th>
<th>Price per minute</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 PM</td>
<td>0.05</td>
</tr>
<tr>
<td>7 PM</td>
<td>0.10</td>
</tr>
<tr>
<td>8 PM</td>
<td>0.25</td>
</tr>
<tr>
<td>9 PM</td>
<td></td>
</tr>
<tr>
<td>10 PM</td>
<td></td>
</tr>
</tbody>
</table>

Peak  Off-peak
This document describes how to charge for usage events based on event attributes in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

■ About Impact Categories
■ About Zone Models
■ About Value Maps
■ About Selectors
■ About Usage Scenario (USC) Selectors
■ About Mapping Services

See also:

■ About Creating Product Offerings
■ Configuring Charge Offers

About Impact Categories

You use impact categories to enable the same charge to apply different pricing based on the values of various event attributes. For example, to configure different pricing for calls made to different countries, you add impact categories for each destination country to the charge. When a call occurs, the pricing associated with the impact category for that call’s destination is applied to the call. For example, to charge usage for a mobile phone service, you might create the following impact categories:

■ Impact Category: Bolivia
  Balance Impact: .10 per minute

■ Impact Category: France
  Balance Impact: .05 per minute

You create an impact category for each zone and usage scenario that you define. A usage scenario describes the type of call, for example, a long-distance mailbox inquiry. For example, you can create impact categories to rate:

■ Local calls.
■ Long distance calls.
■ Local call forwarding.
■ Long distance call forwarding.
Impact categories for zones are used as follows:

- **Base** zone impact categories are used as the results of rules in zone models and in Usage Scenario (USC) selectors.

- **Derived** zone impact categories are used as the results of rules in USC selectors and Access Point Name (APN) selectors. They are considered derived because the rules in USC and APN selectors use additional attributes to remap base zone impact categories to different zone impact categories. For example, a USC selector rule might specify that if the base zone impact category is China and the usage type is **Friends & Family**, the selector returns the Friends & Family derived zone impact category.

You can group multiple impact categories into a single group node to apply the same pricing to all of them.

### About Zone Models

You use zone models to charge for calls based on their origin and destination. BRM supports the following types of zone models:

- **Standard**: A zone model based on the origin and destination numbers of a call. It contains rules that associate a pair of origin and destination numbers with a zone impact category (see "About Impact Categories").

  To specify origin and destination numbers, you must include an international access code (the code used to dial *out of* the country in which the phone number is located). Optionally, you can also include a country code (the code used to dial *in to* the country in which the phone number is located), an area code, a region code, a city code, a phone number prefix, and so on—up to and including the entire phone number.

  Figure 13–1 shows zone rules associated with the same impact category.

*Figure 13–1 Multiple Zone Rules Associated with the Same Impact Category*

<table>
<thead>
<tr>
<th>Service Type</th>
<th>Origin</th>
<th>Destination</th>
<th>Zone Impact Category</th>
<th>Alternate Zone Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>910</td>
<td>530</td>
<td>Local 2</td>
<td>None</td>
</tr>
<tr>
<td>=</td>
<td>408</td>
<td>530</td>
<td>Long Distance 1</td>
<td>None</td>
</tr>
<tr>
<td>=</td>
<td>408</td>
<td>303</td>
<td>Mobile</td>
<td>None</td>
</tr>
<tr>
<td>=</td>
<td>408</td>
<td>415</td>
<td>Mobile</td>
<td>None</td>
</tr>
<tr>
<td>=</td>
<td>510</td>
<td>942</td>
<td>New Zealand</td>
<td>None</td>
</tr>
<tr>
<td>=</td>
<td>525</td>
<td>312</td>
<td>New Zealand</td>
<td>None</td>
</tr>
<tr>
<td>=</td>
<td>512</td>
<td>303</td>
<td>New Zealand</td>
<td>None</td>
</tr>
</tbody>
</table>

- **Geographical**: A zone model based on the distance between the origin and the destination of a call. Geographical zone models include the following:
  - Zone rules that associate a distance with a zone impact category.
  - A list of area codes. Each area code is associated with one or more pairs of longitude and latitude coordinates.

When a customer makes a call, BRM uses the data in the area code list to compute the distance between the origin number and the destination number. BRM then
assigns an impact category to the call event by using the zone rule whose distance most closely matches the computed distance.

Geographical zone models are useful in the following situations:

- Customers are located close to the border between two area codes. For example, if a customer in one area code calls a person two blocks away in another area code, you do not want to charge for a long-distance call.

- The distance covered by an area code is very large and you want to use several rates within the same area code. You do this by associating different pairs of latitude and longitude coordinates with the same area code in a geographical zone model.

When creating charges, you must select a zone model to get a list of impact categories that can be added to the charge. For example, Figure 13–2 shows how the selected zone model determines which impact categories are available.

**Figure 13–2 Using Zone Models to Select Impact Categories**

![Add Impact Categories](image)

**Note:** Optionally, users can also select a USC or APN selector to provide enhanced zone impact categories. See "About Selectors" and "About Impact Categories" for more information.
Alternate Zone Models

In both standard and geographical zone models, each rule results in the application of a zone impact category and, optionally, an alternate zone model (see the rules in Figure 13–1). Alternate zone models enable you to define more granular zone rules and reuse them in multiple zone models.

For example, in a zone model for calls that originate in the U.S., you might include a rule with the following parameters:

- Origin: 011
- Destination: 00
- Impact Category: 00 General
- Alternate Zone Model: US to 00 Country Codes

In this example, the alternate zone model might contain rules for various combinations of calls from the U.S. (011 international access code) to numbers with the 00 international access code and a particular country code, such as 00 44 for calls to the U.K., 00 91 for calls to India, and so on).

When a call is made from the U.S. to a 00 international access code in this example, BRM first checks to see whether the country code of the destination number matches a rule in the alternate zone model. If it does, BRM uses the impact category associated with the rule in the alternate zone model to calculate the call’s charge. If it does not, BRM uses the 00 General impact category to calculate the charge.

See "Creating Pricing Setup Components" for more information about configuring zone models.

Displaying Zoning Information on an Invoice

You can enter a zone description that can be included in invoices. To do so, enter the zone description in the **Description** field in the following dialog boxes:

- Standard Zone
- Geographical Zone
- Usage Scenario Mapping

You can use a maximum of 2,000 characters in the description.

About Value Maps

A value map is a hierarchical structure that associates zone names with values that apply to a single event attribute, such as an area code, an IP address, or a broadband bit rate. A value map can have multiple nested zones.

You use value maps in charge selectors to group event attribute values into manageable categories. For example, to apply the same charge to all calls made from the San Francisco Bay Area to Los Angeles, you might use a California Area Codes value map that includes the following zones:

- Northern California
  - San Francisco Bay Area
    * 408
    * 415
Instead of creating charge selector rules for all possible combinations of those area codes, you associate the California Area Codes value map with the charge selector and then create only one rule in which San Francisco Bay Area is the origin call value and Los Angeles is the destination call value.

**Note:** In PDC, value maps are used only with charge selectors (see "About Selectors").

See "Creating Pricing Setup Components" for information about configuring value maps.

### About Selectors

A selector is a series of rules that associates the values of event attributes, service attributes, account attributes, or custom rules to a result. Different selectors have different types of results. Selectors can be used in charge offers to determine the price of an event.

When an event occurs, the selector rules are evaluated in order of priority. The first rule that applies to the event determines which result is returned by the selector.

You use selectors to enable the same charge offer or discount offer to apply different balance impacts based on various event attributes.

For example, when BRM calculates the cost of a telephone call, a charge selector can determine which charge to use based on call origins and destinations.

You can create the following types of selectors:

- **Charge selector:** Determines the charge to use based on the values of specified event, service, or account attributes. You can use a charge selector instead of a charge in a charge offer.

- **Discount selector:** Determines the discount to use based on the values of specified event, service, or account attributes. You can use a discount selector instead of a discount in a discount offer.

- **Generic selector:** Determines the result name based on the values of specified event attributes, service attributes, or custom rules that is used in a charge offer to guide to a pricing.

- **Price selector:** Determines the appropriate pricing to use based on the values of specified event, service, or account attributes. You can use a price selector instead of a pricing instance in a charge offer.
About Usage Scenario (USC) Selectors

- **Usage Scenario (USC) selector**: Determines a new impact category based on the zone impact category and specific event attributes, such as differentiated network services or friends-and-family calls.

- **Access Point Name (APN) selector**: Determines a new impact category based on the zone impact category and access point name that applies to the event.

When creating a selector, you choose the attributes to be used in the rules. Each rule then uses the same fields, and you must specify a value for all the fields in every rule. For some rules, the value of a field might not be relevant, so you can use a wildcard (*) to indicate that any value is acceptable.

The execution sequence of the rules is important, so you can reorder them as necessary. The rules tables include a search mechanism to make it easy to find a rule.

In all selectors except charge selectors, rules have an effective period. By default, the period starts immediately and never ends. You can modify and add effective periods.

For information about creating selectors, see the PDC Help.

**About Usage Scenario (USC) Selectors**

A Usage Scenario (USC) selector contains rules that map existing impact categories to new impact categories based on usage attributes. For example, If you provide a friends-and-family discount, you can create separate impact categories for national friends and international friends.

You define USC selector rules by specifying values for the existing impact category, the event usage attributes used to evaluate the USC rule, and the new impact category. When an event’s usage attribute values match a rule, the new impact category is returned by the USC selector.

USC selectors bring together the data from service codes, usage classes, and zones, as shown in Figure 13–3.

**Figure 13–3  Usage Scenario Mapping Factors**

The impact category is included as part of the charges.

You define usage scenario mapping to rate differentiated network services, such as mailbox inquiries and friends-and-family discounts.

You can use multiple attributes to define an impact category, or you can use only one attribute.
About Mapping Services

Incoming events from multiple switches often use different codes to represent the same service or supplementary service. To process these events, you must normalize that data by mapping external service codes to internal service codes, service classes, and usage classes.

- An internal service code represents a service such as voice or fax.
- An internal service class represents a variation on the service (for example, a GPRS quality of service (QoS) or a telephone service used only for emergency calls).
- An internal usage class represents a supplementary service (for example, call forwarding or a voice mail box). The data used for rating usage classes comes from the network.

To create internal service mappings, you set up the data that specifies how to map the external data to the internal codes. For example, you can specify that if the external service code is 011, the internal service code is TEL.

You organize service codes, service classes, and usage classes in map groups. A map group specifies the set of services and usage scenarios to be used for rating. You can create any number of map groups.

Mapping Service Codes and Service Classes

To create service code and service class mappings, you specify which external data is used to determine the internal service codes and service classes. For example, you can specify that if the external service code is 011, the internal service code is /service/telco/gsm/telephony.

The mappings are based on the following data:

- External service code
- Usage class
- Location area indicator
- VAS event product code
- Quality of service requested
- Quality of service used
- Record type

Multiple service codes can be mapped to a single BRM service type. For the best performance, use as few service mappings as possible. For example, map the TEL, SMS, DATA, and FAX service codes to a single /service/telco/gsm BRM service type.

Mapping Usage Classes

To create usage class mappings, you set up the data that specifies how to map the external data to the internal usage class. For example, you can specify that if the external supplementary service code for call forwarding is 41, the internal usage code is CW. In addition, you can use various event attributes to create variations on supplementary service for rating purposes.

For example:

- Service code mapping. Use the usage class to create special virtual services and charge offers; for example, you can use the combination of telephony and call forwarding to create value-added services.
- Service class mapping: Use the usage class to create special sub-services or quality of services (for example, service agreements).
- Usage scenario mapping: Use the usage class to map to impact categories.
- Pricing adjustment: Use the usage class to provide discounts or adjustments in individual events.

The mappings are based on the following data:
- External usage class
- Tariff class and subclass
- Record type
- Connect type and Connect subtype, for example, anonymous, from another network, or direct
- Transit area code
- APN address when you use GPRS or UMTS technology
- SS (supplementary service) packet
This document describes how to implement and manage recurring charges in Pricing Design Center (PDC) for Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- About Recurring Charges
- About Cycle Forward Charges
- About Cycle Arrears Charges
- About Cycle Forward Arrears Charges
- About Creating Recurring Charges in PDC
- About Recurring Charges Events

See also:

- Advanced Configuration for Recurring Charges
- About Creating Product Offerings
- Configuring Charge Offers

About Recurring Charges

Recurring charges are ongoing charges that are not generated or affected by usage, such as a charge for a monthly subscription fee. For example:

- A 30-dollar monthly fee for a mobile phone service
- A 5-dollar fee for an automatically renewing smartphone app

Recurring charges do not use Elastic Charging Engine (ECE) for rating. Instead, recurring charges are applied when you run BRM billing utilities.

There are three types of recurring charges:

- **A cycle forward charge** applies a recurring charge in advance. The advantage of using a cycle forward charge is that you collect payments sooner.

- **A cycle arrears charge** applies a recurring charge to the previous month. The advantage of using a cycle arrears charge is that all of the usage charges and recurring charges for a particular month are included in the same bill.

- **A cycle forward arrears charge** applies a recurring charge to the previous month, identical to a cycle arrears charge. The advantage of using a cycle forward arrears charge is that the customer is not charged until the end of the month, but the
About Cycle Forward Charges

To charge a fee in advance, create a charge offer that includes a *cycle forward* charge type, for example, *Monthly Recurring Charge*. You can charge fees in advance at any interval; for example, by using a *Quarterly Recurring Charge* or a *Yearly Recurring Charge* charge type. You can define your own recurring charge types; for example, weekly, or every five months (see "Configuring Flexible Cycle Forward Charges" for more information).

Monthly recurring charges are typically applied at the beginning of an accounting cycle, but they do not need to be synchronized to the start or end of an accounting cycle.

Cycle forward balance impacts are recorded in `/item/cycle_forward` bill items.

About the First Cycle Forward Charge in a New Account

A cycle forward fee incurred at account creation is included in the cycle forward item created when the account is created. Therefore, the first bill includes two cycle forward fees. Figure 14–1 shows an account created on May 15, with the first bill on June 15.

The first cycle forward item includes the charge for the month from May 15 - June 15. The second cycle forward item includes the charge for June 15 - July 15. Both charges are included in the June 15 bill.

![Figure 14–1 Cycle Forward Fees for New Account](image)

About Cycle Arrears Charges

To charge a fee at the end of the month, create a charge offer that includes a *cycle arrears* charge type, for example, *Monthly Recurring Charge Arrear*. Cycle arrears charges apply a charge to the previous billing cycle; the customer pays for the month that has already occurred. Cycle arrears charges are always monthly.

In Figure 14–2, the customer is charged for a cycle forward fee and for a cycle arrears fee. The bill created on April 1 includes:

- A $10 cycle arrears fee for March
- A $10 cycle forward fee for April
About Cycle Forward Arrears Charges

To charge a fee at the end of the month, and recognize unbilled revenue when you run general ledger reports, create a charge offer that includes a cycle forward arrears charge type, for example, Monthly Recurring Charge Forward Arrear.

Just like a cycle arrears charge, a cycle forward arrears charge charges the customer for the previous billing cycle. The difference between cycle arrears charges and cycle forward arrears charges is:

- The balance impact for a cycle arrears charge is applied at the end of the accounting cycle.
- The balance impact for a cycle forward arrears charge is applied at the beginning of the accounting cycle.

However, for both types of charges, the charge is not billed until the end of the billing cycle. Therefore, to the customer, there is no difference in how the charge is billed. The difference appears to your accounting department when you collect general ledger data. Cycle forward arrears charge types allow you to recognize unbilled revenue for the charge when you collect general ledger data before the end of the accounting cycle.

There are no multi-month cycle forward arrears fees.

Cycle forward arrears balance impacts are recorded in /item/cycle_forward_arrear bill items. However, the cycle forward arrears event is assigned to the item that belongs to the next accounting cycle. This way, the fee is tracked in the account balance for the current cycle, but it is not billed until the end of the cycle.

Figure 14–3 shows a cycle forward arrears charge applied on March 1, and billed on April 1. In this case, the balance impact occurs on March 1, but the charge is included in the bill item for April 1.

Figure 14–3  Cycle Forward Arrears Charge
About Implementing Cycle Forward Arrears Charging

Cycle forward arrears events are standard in BRM. However, you must enable delayed billing if you charge for cycle forward arrears events. If you do not otherwise need delayed billing, you can enable delayed billing, but set the delayed billing period to 0.

For information about configuring delayed billing, see “Setting Up Delayed Billing” in BRM Configuring and Running Billing.

About Creating Recurring Charges in PDC

You create a recurring charge in PDC by assigning a recurring charge type to a charge offer. You then apply the same attributes as any charge; for example:

- How the event is measured. Recurring charges are always measured by occurrence.
- When to stop charging; for example, when the associated charge offer is inactivated or closed.
- Tax calculation settings.
- General Ledger (G/L) impacts. You can configure general ledger (G/L) reporting to report earned revenue and unearned revenue. Unearned revenue applies only to cycle forward events and cycle forward arrears events. It never applies to usage events or cycle arrears events.
- Balance impact details, such as the amount to charge, balance element, and so on.

In addition, you can specify when the balance impact of a recurring charge is applied, and whether to prorate recurring charges. See:

- "Specifying When Recurring Charges are Charged"
- "Allowing Recurring Charges to Be Prorated"

---

**Note:** A single charge offer cannot include multiple recurring charges that have the same frequency and type of balance impact, such as a cycle fee. For example, if you add a monthly cycle forward event to a charge offer, you cannot also add a monthly cycle arrears or monthly cycle forward arrears event to the same charge offer. Instead, Oracle recommends that you create separate charge offers for each cycle forward event.

---

Specifying When Recurring Charges are Charged

In Pricing Design Center (PDC), you can configure the balance impact for recurring charges to occur at the following times:

- On the customer’s current billing date. This is the default. If the recurring charge is in a charge offer that is purchased mid-cycle, you can prorate a charge to cover the time between the purchase data and the first billing date.

For example, if the billing date is the 1st of the month and the charge offer is purchased on January 10, the charge is prorated and applied on January 10 for the interval January 10 to February 1. For subsequent cycles, the charge is applied on the billing date (for the interval February 1 to March 1, March 1 to April 1, and so on).

- On the charge offer purchase date. For example, if the billing date is the 1st of the month and the charge offer is purchased on January 10, the charge is applied on
the 10th of every month (for the interval January 10 to February 10, February 10 to March 10, and so on).

- At a specified time before the time period covered by the charge; for example, one month in advance. For example:
  - If you charge a one-month cycle forward fee one month in advance, and the billing day of month (DOM) is the 1st, the charge on May 1st applies to June 1 - June 30. Without in-advance billing, a cycle forward charge on May 1 applies to May 1 - May 31.
  - If you charge a one-month cycle forward fee 15 days in advance, and the billing day of month (DOM) is the 1st, the charge on May 1 applies to May 15 - June 15, the charge on June 1 applies to June 15 - July 15, and so on.

When you charge cycle forward fees in advance, the first cycle forward charge might be larger than the subsequent charges. For example, if a charge offer is purchased on May 1 with in-advance billing of 15 days, the charge on May 1 includes one month in advance (May 15 - June 15) plus 15 days (May 1 - May 14). Subsequent charges are for June 15 - July 15, and so on.

### Allowing Recurring Charges to Be Prorated

You can enable recurring charges to be prorated. A prorated charge typically charges a portion of the fee, based on how long it has been applied. You can also specify to apply no fee, or the full amount. You can specify to prorate charges in the following cases:

- When a charge offer is purchased mid-cycle. This can occur at account creation or when a charge offer is purchase by an existing account.
- When a charge offer that includes a recurring charge is canceled mid-cycle.

If a recurring charge includes more than one balance impact, you can specify which balance impacts should be prorated. For example, if the charge can be prorated, and there are two balance impacts, you can specify to prorate one of the balance impacts but not the other.

Figure 14–4 shows a cycle forward charge applied at account creation on May 1. The billing day of month is May 15, so the cycle forward charge is prorated for the two weeks between account creation and the first billing date.

**Figure 14–4  Prorating Cycle Charges for a Monthly Fee**

<table>
<thead>
<tr>
<th>May 1</th>
<th>May 15</th>
<th>June 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prorated fee</td>
<td>Full monthly fee</td>
<td>Billing date</td>
</tr>
<tr>
<td>Account creation</td>
<td>Billing date</td>
<td></td>
</tr>
</tbody>
</table>

### About Recurring Charges Events

Like all charges, recurring charges are rated by applying a balance impact to an event. Recurring charges are applied to cycle events. The cycle events are:

- `/event/billing/product/fee/cycle/cycle_forward_monthly`
- `/event/billing/product/fee/cycle/cycle_forward_bimonthly`
- `/event/billing/product/fee/cycle/cycle_forward_quarterly`
About Recurring Charges Events

- /event/billing/product/fee/cycle/cycle_forward_semiannual
- /event/billing/product/fee/cycle/cycle_forward_annual
- /event/billing/product/fee/cycle/cycle_arrear
- /event/billing/product/fee/cycle/cycle_forward_arrear

Cycle events are created when you run the pin_cycle_fees billing utility. The pin_cycle_fees utility is typically included in the pin_bill_day script that runs daily.
This document describes advanced functionality for recurring charges in Pricing Design Center (PDC) for Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Configuring Flexible Cycle Forward Charges
- Applying Cycle Forward Fees in Parallel
- Applying Recurring Charges Based on Charge Offer Priority
- Customizing the Cycle Interval for Specified Charge Offers

See also:

- About Creating Product Offerings
- Implementing Recurring Charges
- Configuring Charge Offers

**Configuring Flexible Cycle Forward Charges**

By default, BRM supports monthly, bimonthly, quarterly, semi-annual, and annual cycle forward charges. You can also configure BRM to support *flexible cycle forward charges*. Flexible cycle forward charges can use daily, weekly, monthly, or multimonth cycles that are not restricted to the billing or accounting cycles.

You can use flexible cycle forward charges to grant free balances, provide discounts, or charge fees at any time during the accounting cycle. For example, you can set up a cycle forward fee to grant minutes every week or every day rather than once a month. Or you can set up a monthly cycle forward fee to grant minutes on the 15th of every month, which is different from the monthly accounting cycle that begins the 1st of every month.

Because flexible cycles are not aligned with accounting cycles, cycle forward fees are always prorated.

You set up flexible cycle forward charges by configuring flexible cycle forward events. To set up flexible cycles:

1. Define a custom cycle forward event subclass by using Storable Class Editor in Developer Center.

   For example, to define a cycle forward event that occurs every 10 days, create `/event/billing/product/fee/cycle/cycle_forward_10days`. 
2. Map the new event to a valid purchase level:
   a. In the service-event map configuration file (BRM_home/sys/data/pricing/example/pin_event_map), add an entry for the new cycle forward event. The entry must use the following format:

   purchase_level:event_type:event_description:count: unit

   where:
   count specifies the frequency of the cycle. It must be a positive number.
   unit must be day, week, month, or year.

   For example, to map a biannual (24-month duration) cycle forward event to an account purchase type, the pin_event_map entry is:

   /account:/event/billing/product/fee/cycle/cycle_forward_biannual:Biannual Cycle Forward Event:24:month

   or

   /account:/event/billing/product/fee/cycle/cycle_forward_biannual:Biannual Cycle Forward Event:2:year

   See "Creating Service and Event Storable Classes" in BRM Developer’s Guide.

   b. Run the load_event_map utility. For information on load_event_map, see BRM Setting Up Pricing and Rating.

3. Map the new event type to a valid ratable usage metric (RUM):
   a. In the usage map configuration file, add an entry for the new cycle forward event. For example:

   /event/billing/product/fee/cycle/cycle_forward_biannual:Occurrence: 0: 0: 0: 0: 0: 0: cycle_forward_biannual

   See "Mapping Event Types to RUMs" in BRM Setting Up Pricing and Rating.

   b. Run the load_usage_map utility. For information on load_usage_map, see BRM Setting Up Pricing and Rating.

4. Stop and restart the CM.

5. Use PDC to create charge offers that include your new cycle forward fee.

Charging Cycle Forward Fees Associated with Flexible Cycles

Cycle forward fees are charged when you run monthly billing or by running the pin_cycle_forward billing utility.

When a cycle forward event is generated, balance impacts are applied using balance validity dates.

If a cycle forward event balance impact for a noncurrency balance is set up with a relative cycle start date, balance impacts are applied either to the current cycle or to a future cycle.
Applying Cycle Forward Fees in Parallel

For example, if the relative cycle is set to 1 and the cycle is from 1/1/04 to 3/1/04, a sub-balance is created with a validity period from 1/1/04 to 3/1/04. If the relative cycle is set to 2, a sub-balance is created with a validity period from 3/1/04 to 5/1/04.

Applying Cycle Forward Fees in Parallel

For accounts with multiple services (for example, a wholesale market customer account), you can configure BRM to apply cycle forward fees in parallel for multiple services instead of applying cycle forward fees sequentially for each service, thereby reducing the time to complete billing.

When you configure BRM to apply cycle forward fees in parallel, you can also configure BRM to do the following:

- Enforce cycle fee processing prior to billing. By doing so, BRM eliminates the process of applying cycle forward fees during billing and improves the performance of the billing process.
- Use a single item at the account level to accumulate the cycle charges for all the services. By doing so, BRM reduces the number of items to process and improves overall system performance, which is important when you bill for wholesale customer accounts. When BRM applies cycle forward fees in parallel with service charges aggregated to a single account item, the account can have only a single bill unit (/billinfo object). Even though the account item aggregates the service charges, the respective service balance groups are still updated with the service charges.

Before configuring BRM to apply cycle forward fees in parallel, your configuration must meet the following requirements:

- The number of services attached to a single balance group must be less than 10 in order to get the performance benefit of applying cycle forward fees in parallel.
- There should be no dependency on the order of applying cycle forward fees for account groups (hierarchical, charge sharing, or discount sharing). This is because the cycle forward fees are applied by the pin_cycle_fees utility instead of by the billing application that gives more control on the order of processing accounts in hierarchies.

Applying parallel cycle forward fees involves the following processes:

1. Running the pin_cycle_fees utility in parallel at the services level, which processes cycle forward fees aligned to the accounting cycle.

   **Note:** When the parallel fee processing feature is enabled, cycle forward fees are applied by the pin_cycle_fees utility, which is run before the pin_deferred_act utility. However, when the parallel cycle forward fees processing fees feature is not enabled, cycle forward fees are applied by the billing application after running pin_deferred_act.

2. Running the pin_update_items_journals utility to post-process cycle forward fees.
3. Running the pin_bill_accts utility for regular billing.
4. Running the pin_cycle_fees utility to process flexible cycle forward fees.
5. Running the pin_update_items_journals utility to post-process flexible cycle forward fees.
To configure BRM to apply cycle forward fees in parallel, run the `pin_bus_params` utility to change the `StagedBillingFeeProcessing` business parameter. For information about this utility, see BRM Developer’s Guide.

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   
   ```
   run:
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```

3. Search for the following line:
   
   ```
   <StagedBillingFeeProcessing>0</StagedBillingFeeProcessing>
   ```

   The default is 0. (BRM applies the cycle forward fees as part of the billing process.)

   **Note:** The cycle fee processing at the time of billing is only with reference to the cycle forward fees aligned with the accounting cycle boundary.

4. Do one of the following:
   - To apply cycle forward fees in parallel by service, specify 1.
   - To enforce cycle fee processing prior to billing and apply cycle forward fees in parallel by service, specify 2.
   - To apply cycle forward fees in parallel by service with service charges aggregated to a single account item, specify 3.
   - To enforce cycle fee processing prior to billing and apply cycle forward fees in parallel by service with service charges aggregated to a single account item, specify 4.

5. Save the file as `bus_params_billing.xml`.
6. Load the XML file into the BRM database:
   
   ```
   run:
   pin_bus_params bus_params_billing.xml
   ```

7. Stop and restart the CM.

8. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see BRM System Administrator’s Guide.

### About Enforcing Cycle Forward Fee Processing Prior to Billing

When BRM enforces cycle fee processing prior to billing, the following processes are impacted:

- The `pin_cycle_fees` utility performs additional error processing to set the error status (as needed) on the corresponding bill unit.

   **Important:** If you customize `pin_cycle_fees` and use the application global structure `PIN_FLD_EXTENDED_INFO` provided by the multithreaded application framework to hold custom information at run time, you must consider that `pin_cycle_fees` stores the error processing information in a single array element `PIN_FLD_ERROR_INFO` under `PIN_FLD_EXTENDED_INFO`. 
The billing process aborts if any of the following conditions is true:

- The BILLING_STATUS_FLAGS field of the /billinfo object indicates that there was an error processing one of the cycle forward fees.
- There is at least one service for which cycle fee processing (regular, deferred, deferred purchase, or deferred cancellation) has not been completed for the accounting cycle being billed.

In rare cases, if billing is due for a bill unit for more than one accounting cycles, special handling is required. See "Handling Skipped Billing" for more information.

About Aggregating Service Charges to Account Items

When applying cycle forward fees in parallel by service with service charges aggregated to a single account item, multiple threads of pin_cycle_fees updates a single item. To avoid updating the same item by multiple threads, pin_cycle_fees logs the item and journal updates to temporary tables as follows:

- Logs item updates to the /tmp_events_to_process object in the TMP_EVENTS_TO_PROCESS_T table.
- Logs journal updates to the /tmp_journals_to_process object in the TMP_JOURNALS_TO_PROCESS_T table.

The pin_update_items_journals utility processes the temporary item and journal data and updates the main item and journal tables.

To ensure efficient access of these temporary tables, Oracle recommends the following:

- **Resetting high water mark.** Records are frequently inserted into and deleted from the temporary tables. This can result in fragmentation of the temporary tables. You must reset the high water mark for the temporary tables as the BRM schema user.

  Run the following commands every time before calling the pin_bill_accts inside the pin_bill_day script.

  ```sql
  ALTER TABLE TMP_JOURNALS_TO_PROCESS_T ENABLE ROW MOVEMENT;
  ALTER TABLE TMP_JOURNALS_TO_PROCESS_T SHRINK SPACE;
  ALTER TABLE tmp_events_to_process_t ENABLE ROW MOVEMENT;
  ALTER TABLE tmp_events_to_process_t SHRINK SPACE;
  ```

  For more information about the high water mark, see the Oracle Database documentation.

- **Presetting statistics.** Preset the statistics of the temporary tables that are created during BRM installation by running the following commands as a one-time activity. This enables BRM to avoid a full scan of these tables.

  ```sql
  EXEC dbms_stats.set_table_stats('SCHEMA_NAME','TMP_EVENTS_TO_PROCESS_T','','','200000000,40000000,1250');
  EXEC dbms_stats.set_index_stats('SCHEMA_NAME','I_TMP_EVENTS_ID','','numrows=>200000000,numblkks=>1000000,numdist=>200000000,avglblk=>1,avgdblk=>1,clstfct=>200000000);
  EXEC dbms_stats.set_column_stats('SCHEMA_NAME','TMP_EVENTS_TO_PROCESS_T','POID_ID0','distcnt=>200000000,density=>1/200000000,nullcnt=>0,srec=>srec_eve,avgclen=>11);  
  EXEC dbms_stats.set_table_stats('SCHEMA_NAME','TMP_JOURNALS_TO_PROCESS_T','','','200000000,40000000,1250');
  ```
Applying Cycle Forward Fees in Parallel

Handling Skipped Billing

In rare cases, if billing is due for a bill unit for more than one accounting cycles, special handling is required. This multiple cycle overdue billing is referred to as skipped billing.

For example, consider that the current date is December 1 and BRM did not perform billing for the cycles ending November 1 and October 1. In this case, when you run `pin_bill_day` on the current date, three bills are due to be created.

When BRM tries to calculate the cycle forward fees, the following happens:

- **pin_cycle_fees** applies cycle forward fees due only as of October 1 because October 1 billing has not been processed yet.
- **pin_bill_accts** performs billing only on October 1 and aborts with an error when performing billing on November 1 because cycle forward fees due as of November 1 have not been processed yet.

To handle the case of skipped billing used in this example:

1. Run **pin_cycle_fees**, **pin_update_items_journals**, and **pin_deferred_act** in the following sequence:
   ```
   pin_cycle_fees -defer_purchase
   pin_cycle_fees -defer_cycle_fees
   pin_cycle_fees -defer_cancel
   pin_cycle_fees -regular_cycle_fees
   pin_update_items_journals
   pin_deferred_act
   ```

2. Run **pin_bill_accts**.

3. Repeat step 1 and step 2 twice more, which performs billing for November 1 and December 1.

Using the **pin_bill_day** Script to Apply Parallel Cycle Forward Fees

To support applying cycle forward fees in parallel, the **pin_bill_day** script includes the following commented out sections:

- **Pre-Billing Parallel Cycle Fee Processing**: Includes the following entries for **pin_cycle_fees** and **pin_update_items_journals**:
  ```
  ###### pin_cycle_fees -defer_purchase
  ###### pin_cycle_fees -defer_cycle_fees
  ###### pin_cycle_fees -defer_cancel
  ####### pin_cycle_fees -regular_cycle_fees
  ####### pin_update_items_journals
  ```
Applying Cycle Forward Fees in Parallel

- **Post-Billing Parallel Cycle Fee Processing**: Includes the following entry for `pin_update_items_journals`:
  
  ```bash
  ####### pin_update_items_journals
  ```

  To apply cycle forward fees in parallel by using the `pin_bill_day` script:

  1. Make sure that the `StagedBillingFeeProcessing` parameter is not set to 0.
  2. Open the `BRM_home/bin/pin_bill_day` script in a text editor.
  3. Uncomment the following entries in the Pre-Billing Parallel Cycle Fee Processing section:

  ```bash
  ####### pin_cycle_fees -defer_purchase
  ####### pin_cycle_fees -defer_cycle_fees
  ####### pin_cycle_fees -defer_cancel
  ####### pin_cycle_fees -regular_cycle_fees
  ####### pin_update_items_journals
  ```

  4. Uncomment the following entry in the Post-Billing Parallel Cycle Fee Processing section:

  ```bash
  ####### pin_update_items_journals
  ```

  5. Save and close the file.

  6. Run `pin_bill_day`.

**About Limitations and Impacts of Applying Cycle Forward Fees in Parallel**

This section describes the limitations and impacts of configuring BRM to apply cycle forward fees in parallel:

- In rare cases, when the `pin_cycle_fees` utility successfully creates temporary item and journal data and the subsequent run of the `pin_update_items_journals` utility fails to update the item and journal tables, you must investigate and correct the problem in processing the temporary item and journal data before performing any accounts receivable action or generating ledger reports.

- If the parallel fee processing feature is configured to enforce cycle fee processing before billing, balance impact events that occur before the `pin_cycle_fees` utility runs are aborted with an error.

- If the parallel fee processing feature is not configured to enforce cycle fee processing before billing, balance impact events that occur before the `pin_cycle_fees` utility runs result in triggered billing that can be slow due to serial application of cycle forward fees.

- Performance of the following operations is not improved because parallel cycle fee processing does not apply to these operations:
  - Trial billing
  - Bill Now
  - Billing on purchase
  - Account creation
  - Purchase of bundles
  - Billing time discount
Applying Recurring Charges Based on Charge Offer Priority

When multiple charge offers in a bundle include recurring charges, you can configure BRM to apply recurring charges in the order of charge offer priority. You can apply recurring charges based on charge offer priority during the following operations:

- Bundle purchase or cancellation, for all the charge offers per bundle.
- Billing, for all the charge offers in a bundle per bill unit.

**Important:** This parameter does not prioritize charge offers for recurring charges applied by `pin_cycle_fees -defer_cancel` and does not prioritize customized charge offers.

To enable this feature, run the `pin_bus_params` utility to change the `UsePrioritySubscriptionFees` business parameter. For information about this utility, see BRM Developer’s Guide.

To apply recurring charges based on charge offer priority:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```bash
   pin_bus_params -r BusParamsSubscription bus_params_subscription.xml
   ```
3. In the file, change `disabled` to `enabled`:
   ```xml
   <UsePrioritySubscriptionFees>enabled</UsePrioritySubscriptionFees>
   ```
4. Save the file as `bus_params_subscription.xml`.
5. Load the XML file into the BRM database:
   ```bash
   pin_bus_params bus_params_subscription.xml
   ```
6. Stop and restart the CM.
7. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see BRM System Administrator’s Guide.

Customizing the Cycle Interval for Specified Charge Offers

To customize the time interval for applying cycle forward and cycle arrears fees for a specified charge offer, use the `PCM_OP_SUBSCRIPTION_POL_SPEC_CYCLE_FEE_INTERVAL` policy opcode.

This policy opcode is called by `PCM_OP_SUBSCRIPTION_CYCLE_FORWARD` and `PCM_OP_SUBSCRIPTION_CYCLE_ARREARS`. The type of cycle event is passed in the PIN_FLD_SCOPE field in the input flist.

By default, this policy opcode is an empty hook to facilitate customization of the cycle forward and cycle arrears start (CHARGED_FROM_T) and end dates (CHARGED_TO_T) for a specific charge offer. The start and end dates provided are used to calculate the scale to determine the cycle fee amount to charge or refund.
For example, if a charge offer is purchased on April 1, 2009 with a monthly cycle forward fee of $30 and the purchase, usage, and cycle end dates are set to April 20, 2009, the cycle fee amount is based on the scale for the period April 1, 2009 to April 20, 2009 (20 days) divided by the unit interval from April 1, 2009 to April 30, 2009 (30 days). The cycle fee charged is $20/30 * $30, or $20.

If the charge offer is canceled on April 15, 2009, the refund amount is based on the scale for the period April 15, 2009 to April 20, 2009 (5 days) divided by the unit interval from April 1, 2009 to April 20, 2009 (20 days). Because the refund amount is refunded from the charged amount ($20), the refund is 5/20 * (20/30 * $30), or $5. Here, the scale value 20/30 is the original charge scale or the period the charge offer was valid during the cycle.

The PIN_FLD_SCALE value in the input and output flists is the original charge scale and is used only to calculate the refund scale.

To change the scale (for example if you do not want to refund the full amount), change the start and end dates. The refund scale is calculated based on the dates that you provide.
Customizing the Cycle Interval for Specified Charge Offers
This document describes how to use extended rating attributes (ERAs) in your Oracle Communications Billing and Revenue Management (BRM) system and how BRM charges for events by using ERAs.

Topics in this document:

- About Extended Rating Attributes
- Creating ERAs
- Creating Friends and Family ERAs
- Default ERAs

See also:

- About Creating Product Offerings
- Configuring Charge Offers
- Creating Discount Offers

### About Extended Rating Attributes

Extended rating attributes (ERAs) provide special charges or discounts based on a specific attribute of a service or account, such as a telephone number. For example, you use ERAs to offer special friends and family rates or a birthday discount.

### About Sharing ERAs

You can share the ERA values you configure for one service with other services by using profile sharing groups. For example, you can share the same list of phone numbers for a friends and family ERA among all the customers whose phone numbers are included in the list.

You can share service ERAs only.

For more information, see "Working with Profile Sharing Groups" in *BRM Managing Customers*.

### Creating ERAs

This section is an overview of the main steps you take to create an ERA.
Creating Friends and Family ERAs

BRM has built-in support for friends and family ERAs. This feature enables you to create an ERA with multiple lists, each containing phone numbers, APN addresses, email addresses, or other values. You can also use profile sharing to share a friends and family ERA with multiple services.

Following is a description of the steps for creating a friends and family ERA.

Default ERAs

Table 16–1 summarizes the default account-level ERAs:

<table>
<thead>
<tr>
<th>Function</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assign an account to a business segment for business intelligence reporting.</td>
<td>Business Intelligence Segment ERA</td>
</tr>
<tr>
<td>Assign an account to a customer group, or assign an arbitrary quality level to an account.</td>
<td>Customer type or quality ERA</td>
</tr>
<tr>
<td>Assign an account to a group of accounts that share certain telephony properties.</td>
<td>Corporate agreement ERA</td>
</tr>
<tr>
<td>Create a closed user group, such as a group of all mobile numbers in a company.</td>
<td>Closed user group ERA</td>
</tr>
<tr>
<td>Give a discount for calls made on a specific calendar date.</td>
<td>Special day discount ERA</td>
</tr>
<tr>
<td>Define the home cells for a customer and allow discounts while calling from this area.</td>
<td>Home cell assignment ERA</td>
</tr>
<tr>
<td>Provide discounts to calls made to specific numbers or regions, such as all numbers in a country or area code.</td>
<td>Home region code ERA</td>
</tr>
<tr>
<td>Provide discounts to calls made to specific numbers, such as those for friends and family members.</td>
<td>Friends and family ERA</td>
</tr>
<tr>
<td>Assign a Quality of Service to a service.</td>
<td>Service-level agreement ERA</td>
</tr>
</tbody>
</table>
Configuring Closed User Groups

This document describes how to configure closed user groups for Oracle Communications Billing and Revenue Management (BRM) in Pricing Design Center (PDC).

See also:

- About Creating Product Offerings
- Working with Extended Rating Attributes
- Configuring Charge Offers

Configuring Closed User Groups

You can configure closed user groups for applying special prices for events that originate and terminate between a list of subscribers. For example, you can configure a closed user group for applying special rates on calls between employees of a small company.

To identify the members of a closed user group and apply special prices, you configure a custom rule and add it to a generic selector. The custom rule evaluates the groups to which the originating and terminating subscribers belong. If the originating and terminating subscribers belong to even one common closed user group, the custom rule evaluates as true. The generic selector guides to the price based on how the custom rule is evaluated at runtime.

You can configure closed user groups at the account level or at the service level. To configure closed user groups at the account level, use the OOB_ProfileSpecifications.xml and OOB_CRs.xml files as templates. To configure closed user groups at the service level, use the Sample_ServiceCUG_ProfileSpecification.xml and Sample_ServiceCUG_CR.xml files as templates. See Table 17–1 for the usage and description of each element.

If you configure a closed user group at the account level, associate the closed user group names with the appropriate service identifiers for the account when the customer purchases the charge offer with a closed user group. For example, map the phone numbers to the appropriate closed user group names.

If you configure a closed user group at the service level, configure only the appropriate closed user group names when the customer purchases the charge offer with a closed user group.

Table 17–1 lists the elements in the custom rules XML file, the usage of each element, and a description of how to specify each element in the XML file.
### Table 17–1  Elements in the Custom Rules XML File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>customAnalyzerRules</td>
<td><code>&lt;customAnalyzerRules&gt;</code></td>
<td>The root element of the rule.</td>
</tr>
<tr>
<td>name</td>
<td><code>&lt;name&gt;</code> <code>RuleName</code> <code>&lt;/name&gt;</code></td>
<td><code>RuleName</code> specifies the custom rule name.</td>
</tr>
<tr>
<td>description</td>
<td><code>&lt;description&gt;</code> <code>RuleDescription</code> <code>&lt;/description&gt;</code></td>
<td><code>RuleDescription</code> contains the description of the custom rule.</td>
</tr>
<tr>
<td>priceListName</td>
<td><code>&lt;priceListName&gt;</code> <code>PriceListName</code> <code>&lt;/priceListName&gt;</code></td>
<td><code>PriceListName</code> specifies the price list name. BRM uses only one internal price list. The only value is <code>Default</code>.</td>
</tr>
<tr>
<td>profileSpecName</td>
<td><code>&lt;profileSpecName&gt;</code> <code>ProfileAttributeSpecificationName</code> <code>&lt;/profileSpecName&gt;</code></td>
<td><code>ProfileAttributeSpecificationName</code> specifies the name of the profile attribute specification used for configuring the rule.</td>
</tr>
<tr>
<td>customRuleExpression</td>
<td><code>&lt;customRuleExpression&gt;</code> <code>&lt;dynamicFieldCRExpression&gt;</code> <code>&lt;operator&gt;</code> <code>Operator</code> <code>&lt;operator&gt;</code> <code>&lt;FieldIdentifier&gt;</code> <code>...</code> <code>&lt;/FieldIdentifier&gt;</code> <code>&lt;/dynamicFieldCRExpression&gt;</code> <code>&lt;customRuleExpression&gt;</code></td>
<td>Contains the expression to evaluate closed user groups, where <code>Operator</code> specifies the operator to be used in the expression. The valid value is <code>INTERSECT</code>.</td>
</tr>
</tbody>
</table>
| FieldIdentifier     | `<FieldIdentifier>` `<qualifier>` `Qualifier` `<fieldSelector>` `...` `</fieldSelector>` `</FieldIdentifier>` | Contains the details to identify the fields for evaluating closed user groups, where `Qualifier` specifies the qualifier for identifying the fields. The valid values are:  
  - `ORIGINATING`  
  - `TERMINATING`  
In the `<FieldIdentifier>` element, you can define fields from the service, account, and event attribute specifications. |
| FieldSelector       | `<fieldSelector>` `<operator>` `ConditionOperator` `<operator>` `<selectorFieldName>` `SelectorFieldName` `<selectorFieldKind>` `SelectorFieldKind` `<conditionFieldName>` `ConditionFieldName` `<conditionFieldKind>` `ConditionFieldKind` `</fieldSelector>` | Identifies the closed user group names that match the given product identifier (PUID), for example, a phone number, for both the originating and terminating subscribers.  
**Note:** You can use the `<FieldSelector>` element for configuring account-level closed user groups only.  
where:  
  - `ConditionOperator` specifies the operator to be used in the condition. The valid value is `EQUAL_TO`.  
  - `ConditionFieldName` and `ConditionFieldKind`  
  - `SelectorFieldName` specifies the selector field name. You must define the selector fields based on the condition fields.  
  - `SelectorFieldKind` specifies the type of attributes used for field selection. You can specify any service, event, or account attribute.  
  - `ConditionFieldKind` specifies the condition field name.  
  - `ConditionFieldKind` specifies the type of attributes used for selection. You can specify any service, event, or account attribute. |

---

**Configuring Closed User Groups**

17-2  PDC Creating Product Offerings
This document describes how to configure policy specifications for Oracle Communications Billing and Revenue Management (BRM) in Pricing Design Center (PDC).

See also:
- About Creating Product Offerings
- Configuring Charge Offers

About Configuring Policy Specifications

You configure policy specifications in PDC for policy-driven charging.

Note: Policy specifications can be used only with ECE.

After you configure policy specifications, you can import the policy specifications and policy labels from the customized XML files into the PDC database by running the following command:

```
ImportExportPricing -import -config filename
```

where `filename` is the name of the customized XML file.

The policy specification is imported into the PDC database and transformed to BRM and ECE.

Configuring Policy Specifications

To configure policy specifications:

1. Make a copy of the `PDC_home/apps/Samples/Examples/SamplePolicyLabel.xml` file to customize.
2. Open the file in a text editor or XML editor.
3. Add or modify the elements as required.
4. Save and close the file.
5. Make a copy of the `PDC_home/apps/Samples/Examples/SamplePolicySpecification.xml` file to customize.
6. Open the file in a text editor or XML editor.
7. Add or modify the elements as required.
8. Save and close the file.
9. Go to the directory where you have saved the files.

10. Import the policy specifications and policy labels from the customized XML files into the PDC database by running the following command:

    ImportExportPricing -import -config filename

    where filename is the name of the customized XML file.

The policy specifications and policy labels are imported into the PDC database and transformed to ECE.

After you configure policy specifications, you can export the policy specifications and policy labels in PDC to XML files by running the following command:

    ImportExportPricing -export -config POLICY_SPECIFICATION
    ImportExportPricing -export -config POLICY_LABEL

You can load the updated sample XML files into PDC by using the ImportExportPricing utility. See "ImportExportPricing" for the utility’s syntax and parameter descriptions.
This part describes how to manage balances using Oracle Communications Billing and Revenue Management (BRM) pricing components and setup components created in Pricing Design Center (PDC).

Part III contains the following chapters:

- Managing Balances during Charging
- Managing Sub-Balances
- Configuring Item Type Selectors
- About Folds
- Applying Credit Limits to Balances
- About Rollovers
- Configuring Balance Impact Rounding
Managing Balances during Charging

This document describes how balances are managed in Oracle Communications Billing and Revenue Management (BRM) product offerings created using Pricing Design Center (PDC).

See also:
- Managing Sub-Balances
- Configuring Item Type Selectors
- About Folds
- Applying Credit Limits to Balances
- About Rollovers
- Configuring Balance Impact Rounding
- About Creating Product Offerings
- Configuring Charge Offers

Tracking Balances in Balance Groups

Using a service can affect multiple types of balances. For example, a phone service might be charged in both dollars and minutes. In this case, the service usage would be associated with a balance group that includes two balances, one for dollars and one for minutes. Balance groups are collections of multiple balances that apply to one or more services.

You create balance groups when you create packages in PDC. By default, a package contains one balance group: the account balance group. Balances for every service in the package share that balance group, which means that balances such as included minutes are shared among all services. Figure 19–1 shows a default balance group with two types of balances, both of which are used by two services.
To track and control the allocation and consumption of balances for specific services, you can create multiple balance groups and assign services to their own balance groups, or group sets of services by balance group. Figure 19–2 shows two services that each use their own balance group.

In addition to creating different balance groups for different services, you can create different balance groups for different packages that are purchased by the same account. For example, consider a family of four that has a mobile phone service for each family member. The bundle for each service includes 300 minutes. If the package has only the default account balance group, all four family members’ included minutes go into the same account balance, which would contain 1200 included minutes. One teenager could then use 1000 minutes, leaving only 200 minutes to be shared by the other three family members.

To avoid that situation, you could create a package with two mobile phone services in the same bundle for the parents and an optional add-on package in its own balance group. When the parents purchase add-on packages for the children, each child’s included minutes are tracked separately in a separate balance group. Figure 19–3 shows two balance groups, created by purchasing separate packages in the same account.
You can set different credit limits for the same balance element in each balance group. For example, the parents might request a credit limit of $10 on the US Dollar balance element in each child’s balance group to control overage charges but request a $100 credit limit on the US Dollar balance in the balance group for their mobile phone services. See "Applying Credit Limits to Balances" for more information.
Managing Sub-Balances

This document describes how to manage sub-balances in Oracle Communications Billing and Revenue Management product offerings. It describes balance impacts, credit limits, and rollovers.

Topics in this document:

- About Sub-Balances
- About Noncurrency Sub-Balances
- How Balances in Validity-Based Sub-Balances Are Updated
- Restricting the Validity of Noncurrency Sub-Balances That Start on First Usage
- Configuring Custom Sub-Balances
- Specifying the Order in Which Sub-Balances Are Consumed
- Configuring Time-Stamp Rounding for Cycle Events
- Configuring Time-Stamp Rounding for Purchase Events

See also:

- Managing Balances during Charging
- About Rollovers
- About Creating Product Offerings

About Sub-Balances

Each balance in a balance group can include one or more sub-balances. A balance includes sub-balances when portions of the balance are valid at different times. For example, a balance of minutes might include 300 minutes that are valid only for the current month and 1000 minutes that never expire. Figure 20–1 shows an account with two balance groups, one for each service. In the balance group for wireless minutes, there are balances with different expiration dates. Therefore, there are two sub-balances for wireless minutes.
A currency sub-balance can store the balances for multiple services. For example, an account that owns two charge offers that cost $25.00 per offer has a starting currency sub-balance of $50.00, providing the services are associated with the same balance group and have the same validity period.

Account sub-balances include the following information:

- The start time and end time for which the sub-balance is valid.
  Balances with the same validity periods are stored in the same sub-balance. Balances with unique validity periods are stored in separate sub-balances. For more information, see "How Balances in Validity-Based Sub-Balances Are Updated".

- The current amount of the sub-balance.

- The fields in the event record or object (referred to as “contributors”) that contribute to how sub-balances are created, updated, and retrieved. For example, to retrieve the total available balance for a specific service, the service object is specified. To deduct minutes for a phone call, the session object is specified. A separate sub-balance is kept for each unique contributor.

- Rollover data such as the rollover period and the balance amount that is rolled over, if any. See "About Rollovers".

- ID of the offer that granted the balance (referred to as the “grantor object.”)

If the timestamp_rounding entry in the CM pin.conf file is enabled, the start time of the balances granted by cycle events is rounded to midnight. However, the start time of the balances granted by purchase events is not rounded to midnight.

You can configure sub-balances to track various types of balances and usage. See "Configuring Custom Sub-Balances".

**About Noncurrency Sub-Balances**

A noncurrency sub-balance typically has a limited validity period (for example, the period during which minutes can be used). Noncurrency sub-balances can contain various types of balances, such as the following:

- Minutes
- Frequent flyer miles
- Loyalty points
Number of emails or text messages

A noncurrency sub-balance can also keep track of the total balances used for discounts that are shared among several accounts. In this case, the sub-balance acts as a counter to keep track of the total consumed balance.

When granting a noncurrency balance, if a sub-balance already exists, BRM compares the new balance data with the following data in the existing valid sub-balances:

- Contributor
- Grantor object
- Rollover data
- Valid-from date
- Valid-to date

If the data matches, BRM adds the amount to the existing sub-balance; otherwise, it creates a new sub-balance.

For information about configuring sub-balances, see "Configuring Custom Sub-Balances".

How Balances in Validity-Based Sub-Balances Are Updated

By default, BRM stores balances with the same validity periods in the same sub-balance, provided they are associated with the same balance group. BRM automatically creates a new sub-balance for balances with a unique validity period, if one does not already exist.

For example, an account owns two services that each include 100 minutes that are always valid. The account has a balance of 200 minutes stored in a single sub-balance. When the customer uses minutes from each service, the minutes are consumed from the common sub-balance.

Balances with different validity periods are tracked in separate sub-balances. For example, an account owns two services that each include 100 minutes. Minutes for service 1 expire at the end of the month, and minutes for service 2 expire at the end of the year. Each set of minutes is stored in a separate sub-balance.

Note: When noncurrency balances are configured to start on first usage, BRM creates a new sub-balance for each balance whether or not they have the same validity period. See "Restricting the Validity of Noncurrency Sub-Balances That Start on First Usage".

You can specify the order in which sub-balances are consumed by setting up balance consumption rules. See "Specifying the Order in Which Sub-Balances Are Consumed".

You can also limit how validity-based balances such as minutes are summed by configuring sub-balances. For example, you might want to limit usage of minutes to a specific service or a specific call session. See "Configuring Custom Sub-Balances".

You can configure BRM to round time stamps to midnight for balances granted by cycle and purchase events. See the following:

- Configuring Time-Stamp Rounding for Cycle Events
- Configuring Time-Stamp Rounding for Purchase Events
For more information, see "About Configuring Time-Stamp Rounding" in BRM Configuring and Running Billing.

Restricting the Validity of Noncurrency Sub-Balances That Start on First Usage

When a noncurrency balance is configured to start on first usage (when the customer first uses the service), BRM always creates a new sub-balance for that balance when it is granted. A new sub-balance is created for each balance even when a charge offer or discount offer grants multiple first-usage balances of the same type. A balance that has a first-usage start time will remain available for consumption for as long as the balance is not used.

Note: If a balance is shared among accounts in a sharing group, the balance validity period is set when any account in the group first impacts the balance. Because the same balance is shared with all accounts in the group, the validity period of that balance applies to all accounts.

You can configure BRM to automatically restrict the validity period of granted noncurrency balances to end no later than the end time of the charge offer or discount offer that grants the balance. Restricting the balance end time to the offer’s end time ensures that the balance cannot continue to be consumed after the offer expires.

Note: When an offer is canceled, the validity period end time of balances granted by that charge offer or discount offer is set to the time of the cancellation.

To enable this feature, run the `pin_bus_params` utility to change the `RestrictResourceValidityToOffer` business parameter. For information about this utility, see BRM Developer’s Guide.

To restrict balance validity end times to charge offer or discount offer end times:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```
   pin_bus_params -r BusParamsMultiBal bus_params_multi_bal.xml
   
   In the file, change FALSE to TRUE
   ```

   ```
   <RestrictResourceValidityToOffer>TRUE</RestrictResourceValidityToOffer>
   ```

3. Save the file as `bus_params_multi_bal.xml`.
4. Load the XML file into the BRM database:
   ```
   pin_bus_params bus_params_multi_bal.xml
   ```
5. Stop and restart the CM.
6. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see BRM System Administrator’s Guide.
Configuring Custom Sub-Balances

As described in "About Sub-Balances," BRM automatically creates sub-balances when portions of the balance are valid at different times. You do not need to configure sub-balances based on validity times.

You can configure custom sub-balances to track balances based on other values. For example:

- Minutes per call session
- Frequent flyer miles per service instance
- Friends and family calls to specific locations

You configure custom sub-balances by editing and loading a configuration file.

To customize the allocation and consumption of balances, you configure sub-balances by specifying the following values:

- **Balance element ID.** This is the ID for the type of balance in the sub-balance, such as dollars or minutes.
- **Event type.** This is the type of event that impacts the sub-balance, such as GSM usage events (/event/session/telco/gsm).
- **Contributors.** Contributors can be any field in the event record, or from an object related to the charge, such as a service object or account object.

For example, you might configure a custom sub-balance to track balance impacts for:

- Dollars
- Charged by GPRS events
- Tracked by each GPRS session

Sub-balance contributors define how to track sub-balances. For example, if the contributor is a service object, sub-balances are tracked separately for each service instance.

Sub-balance contributors are specified by a field name from the event, or from an object related to the charge, such as a service object or account object. Some fields you might want to use as contributors include:

- **The service object.**
  Specify a service field to track sub-balances for specific service instances such as fax, telephony, and text messaging.

**Note:** Specifying the service object in the configuration is one way of creating service balances. The other way is to create a balance group for the service when you create your packages in PDC.

- **The session object.**
  Specify a session field to track balances per session instance. This is useful when you offer discounts for certain levels of usage (for example, 10 frequent flyer miles per hour of phone calls).
- **The charge offer object.**
When several charge offers in the same package have different rollover rules, tracking balances per charge offer permits BRM to update the minutes for each charge offer.

- The phone number field.
  Specify a phone number field to track balances for called telephone numbers.

- The account object.
  Specify the account field to track balances that are shared among several accounts, such as earned minutes. When minutes are distributed, they can be divided based on each account’s usage level.

There are two types of contributors:

- **Retrieving contributor.** This is the field from which BRM retrieves the amount to add to the sub-balance. For example, if the retrieving contributor is the service for a GPRS event, a separate balance summary is retrieved from each unique GPRS service object.

  All sub-balances with common retrieving contributors are summed when sub-balances are retrieved. For example, to retrieve the total balance for specific services, specify the service object as the retrieving contributor.

- **Updating contributor.** Whereas the retrieving contributor determines the balance amount to retrieve from a charge, the updating contributor specifies how to apply the amount to a sub-balance. For example, to add or deduct minutes for specific broadband sessions, specify the session object as the updating contributor.

Table 20–1 shows contributor configurations to track balances for GSM services; in this case, one sub-balance for dollars and one sub-balance for minutes.

<table>
<thead>
<tr>
<th>Balance Element</th>
<th>Event Type</th>
<th>Retrieving Contributor</th>
<th>Updating Contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dollars</td>
<td>/event/session/gsm</td>
<td>PIN_FLD_SERVICE_OBJ</td>
<td>PIN_FLD_SERVICE_OBJ</td>
</tr>
<tr>
<td>Minutes</td>
<td>/event/session/gsm</td>
<td>PIN_FLD_SERVICE_OBJ</td>
<td>PIN_FLD_SERVICE_OBJ</td>
</tr>
</tbody>
</table>

Each row in this example specifies a sub-balance:

- **Dollars balance impact**
  - **Dollars** is a currency sub-balance that is created in the account balance group.
  - **/event/session/gsm** is the event type that will impact the dollars sub-balance.
  - **PIN_FLD_SERVICE_OBJ** is the retrieving contributor, so the dollars balance for each unique GSM service is summed when retrieving balance information.

  If you specify an asterisk instead of PIN_FLD_SERVICE_OBJ for the retrieving contributor, dollars for all GSM services owned by the account would be summed when retrieving balance information.

  - **PIN_FLD_SERVICE_OBJ** is the updating contributor, so a separate sub-balance is created to track dollars for every unique GSM service that the account owns, such as telephony, fax, and SMS.

- **Minutes balance impact**
- **Minutes** is a noncurrency sub-balance that is created in the account balance group.
- `/event/session/gsm` is the event type that will impact the minutes balance.
- `PIN_FLD_SERVICE_OBJ` is the retrieving contributor, so the minutes balance for each unique GSM service is summed when retrieving balance information.
  
  If you specify an asterisk instead of `PIN_FLD_SERVICE_OBJ` for the retrieving contributor, minutes for *all* GSM services owned by the account would be summed when retrieving balance information.

- `PIN_FLD_SERVICE_OBJ` is the updating contributor, so a separate sub-balance is created to track minutes for every unique GSM service that includes minutes. Additional minutes and consumption of minutes are restricted to the GSM service instance.

When a balance impact affects more than one sub-balance, the sub-balances are updated in the following order:

1. **By validity period.** By default, sub-balances are updated in chronological order based on the validity start date or end date. You specify the order in which sub-balances are used by setting up balance consumption rules. See "Specifying the Order in Which Sub-Balances Are Consumed".

2. **By contributor.** Sub-balances with specific field contributors are impacted before sub-balances that accept all contributors (with an asterisk or empty contributor values). See "Configuring Custom Sub-Balances".

## Defining and Loading Custom Sub-Balances

See "Configuring Custom Sub-Balances," for information.

To configure sub-balances, you edit the sub-balance configuration file and then run the `load_pin_sub_bal_contributor` utility to load the contents of the file into the `/config/sub_bal_contributor` object in the BRM database.

Each line in the `pin_sub_bal_contributor` file defines the usage for which a sub-balance is created. The configurations apply to all sub-balances in the BRM database, but they are implemented at the balance group level. That is, when a usage event occurs, if the account has more than one balance group, only the sub-balances within the specified balance groups are impacted.

To add sub-balance configurations, use this syntax:

```
balance_type:event_type:retrieving_contributor:updating_contributor
```

For example:

```
1000003:/event/session/gprs:PIN_FLD_SESSION_OBJ:PIN_FLD_SESSION_OBJ
```

where:

- **1000003** is the balance element ID for frequent flyer miles.
- **/event/session/gprs** is the event type that impacts the frequent flyer miles balance.
- `PIN_FLD_SESSION_OBJ` is the retrieving contributor. Therefore, a separate balance summary is retrieved for each unique GPRS session.
- `PIN_FLD_SESSION_OBJ` is the updating contributor. Therefore, separate sub-balances are created or updated for each unique GPRS session.
Specifying the Order in Which Sub-Balances Are Consumed

The contributors can be a specific field or an asterisk (*) to indicate any contributor. For example, the following entry retrieves and tracks dollars (balance element type 840) in a single sub-balance for all events with any contributor:

```
840 : /event:*:* *
```

To configure several sub-balances for one balance element, leave subsequent balance element fields empty. If no balance element is specified, the previous balance element specified is used. For example, the next two configurations use the same balance element (100002):

```
100002 : /event/session/telco/gsm : PIN_FLD_SESSION_OBJ : PIN_FLD_SESSION_OBJ
       : /event/session/call : PIN_FLD_SERVICE_OBJ : PIN_FLD_SERVICE_OBJ
```

After editing the file, you load it into the BRM database.

---

**Caution:** When you run the `load_pin_sub_bal_contributor` utility, it replaces the existing sub-balance configurations. If you are updating a set of sub-balance configurations, you cannot load new configurations only. You load complete sets of sub-balance configurations each time you run the `load_pin_sub_bal_contributor` utility.

---

Use the following command to run the `load_pin_sub_bal_contributor` utility:

```
load_pin_sub_bal_contributor pin_sub_bal_contributor
```

If you are not in the same directory as the `pin_sub_bal_contributor` file, include the complete path to the file. For example:

```
load_pin_sub_bal_contributor BRM_home/sys/data/pricing/example/pin_sub_bal_contributor
```

For more information, see "load_pin_sub_bal_contributor".

To verify that the sub-balance configurations loaded correctly, display the `/config/sub_bal_contributor` object by using Object Browser, or use the `robj` command with the `testnap` utility.

Specifying the Order in Which Sub-Balances Are Consumed

In a balance group, balances for the same type of balance element are combined into a sub-balance. For example, an account that owns two charge offers that each include 300 minutes has a starting included minute balance of 600 minutes, providing the services are associated with the same balance group and have the same validity period for the included minutes.

When portions of a balance are valid at different times, however, BRM creates multiple sub-balances for that balance in the balance group. For example, a balance group might include 300 minutes that are valid only for the current billing cycle and 1000 minutes that never expire. Because the included minutes have different validity periods, they are tracked in different sub-balances.

When a customer uses a service, BRM must know which sub-balance to use first. To specify the order in which sub-balances are consumed, you use consumption rules. The rules are based on the start and end times of the sub-balances, for example consume the sub-balance with the earliest validity start time first.
For example, you can specify whether a subscriber with the following minute sub-balances uses the Anytime Minutes or the rollover Anytime Minutes first:

- 100 Anytime Minutes that are valid March 1 to April 30
- 50 rollover Anytime Minutes that are valid February 1 to March 30

If the minute balance is associated with a rule that says to consume the sub-balance that expires first, the 50 rollover Anytime Minutes are used first.

If the minute balance is associated with a rule that says to consume the sub-balance with the latest start time, the 100 Anytime Minutes are used first.

BRM supports the balance consumption rules shown in **Table 20–2**:

**Table 20–2  Supported Consumption Rules**

<table>
<thead>
<tr>
<th>Consumption Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earliest start time (EST)</td>
<td>Consume the sub-balance with the earliest validity start time first.</td>
</tr>
<tr>
<td>Latest start time (LST)</td>
<td>Consume the sub-balance with the latest validity start time first.</td>
</tr>
<tr>
<td>Earliest expiration time (EET)</td>
<td>Consume the sub-balance with the earliest validity end time first.</td>
</tr>
<tr>
<td>Latest expiration time (LET)</td>
<td>Consume the sub-balance with the latest validity end time first.</td>
</tr>
<tr>
<td>Earliest start time and latest expiration time (ESTLET)</td>
<td>Consume the sub-balance with the earliest validity start time first. If multiple sub-balances have the same start time, use the one with the latest end time first.</td>
</tr>
<tr>
<td>Earliest start time and earliest expiration time (ESTEET)</td>
<td>Consume the sub-balance with the earliest validity start time first. If multiple sub-balances have the same start time, use the one with the earliest validity end time first.</td>
</tr>
<tr>
<td>Latest start time and earliest expiration time (LSTEET)</td>
<td>Consume the sub-balance with the latest validity start time first. If multiple sub-balances have the same validity start time, use the one with the earliest validity end time first.</td>
</tr>
<tr>
<td>Latest start time and latest expiration time (LSTLET)</td>
<td>Consume the sub-balance with the latest validity start time first. If multiple sub-balances have the same validity start time, use the one with the latest validity end time first.</td>
</tr>
<tr>
<td>Earliest expiration time and earliest start time (EETEST)</td>
<td>Consume the sub-balance with the earliest validity end time first. If multiple sub-balances have the same validity end time, use the one with the earliest validity start time first.</td>
</tr>
<tr>
<td>Earliest expiration time and latest start time (EETLST)</td>
<td>Consume the sub-balance with the earliest validity end time first. If multiple sub-balances have the same validity end time, use the one with the earliest validity start time first.</td>
</tr>
<tr>
<td>Latest expiration time and earliest start time (LETEST)</td>
<td>Consume the sub-balance with the latest validity end time first. If multiple sub-balances have the same validity end time, use the one with the earliest validity start time first.</td>
</tr>
<tr>
<td>Latest expiration time and latest start time (LETLST)</td>
<td>Consume the sub-balance with the latest validity end time first. If multiple sub-balances have the same validity end time, use the one with the latest validity start time first.</td>
</tr>
</tbody>
</table>

As shown in **Figure 20–2**, you associate consumption rules with balances in packages.
You can also set systemwide and default balance element settings. BRM reads and uses the consumption rule settings in the order shown below:

1. **Package consumption rules.** Your product offerings can include consumption rules for each service that you support, defined differently in each package. This enables you to have different consumption rules for the same balance based in the packages purchased by the customer. When a customer purchases a package, the package’s balance element-to-consumption rule mapping is stored in the customer’s /balance_group object. If there are any conflicting rules for the same balance element, BRM uses the rule from the most recently purchased package.

2. **Systemwide service consumption rules.** You can specify a systemwide consumption rule for each service that you support. BRM uses the systemwide settings only if a rule is not defined for the balance element in the customer’s purchased packages. BRM stores systemwide settings in the /config/beid object.

   You define systemwide consumption rules for each balance element that you support by using BDC. See BDC Help for more information.

3. **Default consumption rule.** You can specify a default consumption rule that applies to all balance elements. This setting is used only if a consumption rule is not defined for the balance element in the customer’s purchased packages or if there is not a systemwide service rule. BRM stores the default balance element setting in the /config/business_params object.

   You define the default consumption rule in the /config/business_params object. See “Setting the Default Consumption Rule” for more information.

### Setting the Default Consumption Rule

The default consumption rule applies to all balance elements in your system. The default setting is earliest start time and earliest expiration time (ESTEET). You can change this setting by running the `pin_bus_params` utility to change the `SortValidityBy` business parameter. For information about this utility, see BRM Developer’s Guide.

To set the default balance element consumption rule:

1. Go to `BRM_home/sys/data/config`.

2. Create an XML file from the `/config/business_params` object:

   ```bash
   pin_bus_params -r BusParamsMultiBal bus_params_multi_bal.xml
   ```
3. In the file, change `ESTEET` to one of the values listed in Table 20–2:
   `<SortValidityBy>ESTEET</SortValidityBy>`

4. Save the file as `bus_params_multi_bal.xml`.

5. Load the XML file into the BRM database:
   ```
   pin_bus_params bus_params_multi_bal.xml
   ```

6. Stop and restart the CM.

7. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see `BRM System Administrator’s Guide`.

### Configuring the Default Consumption Rule in ECE

When ECE receives a usage request for which no consumption rules are configured, ECE applies its own systemwide default consumption rule for processing the usage request.

To configure the default consumption rule in ECE:

1. Access the ECE configuration MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Connect to the ECE charging server node enabled for JMX management.
      This is the charging server node set to `start CohMgt = true` in the `ECE_home/occeserver/config/eceTopology.conf` file.
   d. Start a JMX editor that enables you to edit MBean attributes, such as JConsole.
   e. In the editor’s MBean hierarchy, find the ECE configuration MBeans.

2. Expand the `ECE Configuration` node.

3. Expand `charging.server`.

4. Expand `Attributes`.

5. Set the `systemConsumptionRule` attribute to one of the following systemwide consumption rules:
   - `EARLIEST_START`
   - `LATEST_START`
   - `EARLIEST_Expiration`
   - `LATEST_Expiration`
   - `EARLIEST_START_LATEST_Expiration`
   - `EARLIEST_START_EARLIEST_Expiration`
   - `LATEST_START_LATEST_Expiration`
   - `LATEST_START_EARLIEST_Expiration`
   - `EARLIEST_Expiration_EARLIEST_START`
   - `EARLIEST_Expiration_LATEST_START`
   - `LATEST_Expiration_EARLIEST_START`
   - `LATEST_Expiration_LATEST_START`
Configuring Time-Stamp Rounding for Cycle Events

- **LATEST_EXPIRATION_LATEST_START**
- **NONE**: When the attribute is set to NONE, the default consumption rule is not configured, and the order for consuming balances is undefined.

By default, this attribute is set to **EARLIEST_START_EARLIEST_EXPIRATION**.

6. Save your changes.

**Deleting Expired Sub-Balances**

To delete expired sub-balances, use the `pin_sub_balance_cleanup` utility.

**Configuring Time-Stamp Rounding for Cycle Events**

You can configure BRM to round time stamps to midnight for balances granted by cycle events. For more information, see "About Configuring Time-Stamp Rounding" in BRM Configuring and Running Billing.

To configure BRM to round time stamps to midnight for the resources granted by cycle events:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`) in a text editor.
2. Set the `timestamp_rounding` entry to 1.
3. Save and close the file.

**Configuring Time-Stamp Rounding for Purchase Events**

You can configure BRM to round time stamps to midnight for balances granted by purchase events. For more information, see "About Configuring Time-Stamp Rounding" in BRM Configuring and Running Billing.

To enable this feature, run the `pin_bus_params` utility to change the `TimestampRoundingForPurchaseGrant` business parameter. For information about this utility, see BRM Developer’s Guide.

To configure BRM to round time stamps to midnight for the resources granted by purchase events:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`) in a text editor.
2. Set the `timestamp_rounding` entry to 1.
3. Save and close the file.
4. Go to `BRM_home/sys/data/config`.
5. Create an XML file from the `/config/business_params` object:
   ```bash
   pin_bus_params -r BusParamsRating bus_params_rating.xml
   ```
6. In the file, change `disabled` to `enabled`:
   ```xml
   <TimestampRoundingForPurchaseGrant>enabled</TimestampRoundingForPurchaseGrant>
   ```
7. Save the file as `bus_params_rating.xml`.
8. Load the XML file into the BRM database:
   ```bash
   pin_bus_params bus_params_rating.xml
   ```
9. Stop and restart the CM.
10. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see *BRM System Administrator’s Guide*. 
Configuring Time-Stamp Rounding for Purchase Events
This document describes how to configure item type selectors in Pricing Design Center (PDC) to track balances in Oracle Communications Billing and Revenue Management product offerings.

Topics in this document:

- Configuring Item Type Selectors
- Loading the OOB_ItemTypeSelector.XML File
- Configuring Item Type Selectors

See also:

- Managing Balances during Charging
- Managing Sub-Balances
- About Creating Product Offerings

### Configuring Item Type Selectors

You configure item type selectors in PDC so that balance impacts can be tracked appropriately for different bill items.

By default, BRM tracks balances for the following bill items: cycle arrears items, cycle forward items, cycle forward arrears items, cycle tax items, cycle incentive items, and usage items. You can load the OOB_ItemTypeSelector.xml file into PDC to configure the item type selectors for these bill items in PDC. See "Loading the OOB_ItemTypeSelector.XML File" for more information.

If you want to use custom bill items, you can configure item type selectors to create rules to assign balance impact to the custom bill items. You can use the Sample_ItemTypeSelector.xml file as a template to configure the item type selectors for custom bill items. See "Configuring Item Type Selectors" for more information.

**Table 21–1** lists the elements that are supported in the item type selector XML file, the usage of each element, and a description of how to specify each element in the XML file.
Table 21–1  Elements Supported in the Item Type Selector XML File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>itemTypeSelectors</td>
<td><code>&lt;itemTypeSelectors&gt;</code></td>
<td>The root element of <code>ItemTypeSelector.xml</code>.</td>
</tr>
<tr>
<td>name</td>
<td><code>&lt;name&gt;ItemTypeSelectorName&lt;/name&gt;</code></td>
<td><code>ItemTypeSelectorName</code> specifies the item type selector name.</td>
</tr>
<tr>
<td>description</td>
<td><code>&lt;description&gt;ItemTypeSelectorDescription&lt;/description&gt;</code></td>
<td><code>ItemTypeSelectorDescription</code> contains the description of the item type selector.</td>
</tr>
<tr>
<td>priceListName</td>
<td><code>&lt;priceListName&gt;PriceListName&lt;/priceListName&gt;</code></td>
<td><code>PriceListName</code> specifies the price list name. <code>BRM</code> uses only one internal price list. The only value is <code>Default</code>.</td>
</tr>
<tr>
<td>applicableToName</td>
<td><code>&lt;applicableToName&gt;ApplicableToName&lt;/applicableToName&gt;</code></td>
<td><code>ApplicableToName</code> specifies the service name or <code>Account</code>. The attributes of this service or account are used in the rules.</td>
</tr>
<tr>
<td>eventSpecName</td>
<td><code>&lt;eventSpecName&gt;EventName&lt;/eventSpecName&gt;</code></td>
<td><code>EventName</code> specifies the name of any event. The attributes of this event are used in the rules.</td>
</tr>
<tr>
<td>itemGroup</td>
<td><code>&lt;itemGroup&gt;ItemGroupName&lt;/itemGroup&gt;</code></td>
<td>Specifies the item group to be used for aggregating the balance impacts; where <code>ItemGroupName</code> specifies the name of the item group. The <code>&lt;itemGroup&gt;</code> element is associated with the <code>&lt;ItemTag&gt;</code> element in <code>BRM</code>. To aggregate balance impacts based on the item configuration value in the business profile, specify the item configuration value as <code>ItemGroupName</code>. See the discussion about improving performance by using multiple item configurations in <code>BRM System Administrator's Guide</code> for more information on item configuration. <strong>Note</strong>: If you specify the item configuration value as <code>ItemGroupName</code>, Oracle recommends that you do not change that value in the business profile. The default value is <code>Default</code>.</td>
</tr>
</tbody>
</table>
| applicableToAllBalImpact | `<applicableToAllBalImpact>BalanceImpactsOption</applicableToAllBalImpact>` | Specifies whether this item type selector is applicable to all balance impacts, where `BalanceImpactsOption` is:  
  - `true` if the item type selector applies for all the balance impacts  
  - `false` if the item type selector applies only to specific balance impacts |
Contains the rules for assigning the balance impacts to the bill items, where:

- **RuleName** specifies the rule name.
- **RuleOrder** specifies the order in which the rules must be applied.
- **Expression** contains the details to derive which items to assign balance impacts to.

**Note:** You can add multiple rules in the item type selector. Each rule points to the item specification in the same item type selector.

Each rule in the item type selector can refer to a different attribute. For example, Rule 1 can use an event attribute, Rule 2 can use an attribute of the charging result, and Rule 3 can use a service attribute.
### Configuring Item Type Selectors

Table 21–1  (Cont.)  Elements Supported in the Item Type Selector XML File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>expression</td>
<td>&lt;expression&gt; &lt;separator&gt;&lt;operator&gt; &lt;fieldName&gt;&lt;fieldKind&gt;&lt;fieldValue&gt;</td>
<td>Contains the details to derive which items to assign balance impacts to.</td>
</tr>
<tr>
<td></td>
<td>&lt;/fieldValue&gt;&lt;/fieldKind&gt;&lt;/fieldName&gt;&lt;/operator&gt;&lt;/separator&gt;&lt;/expression&gt;</td>
<td>- <strong>Delimiter</strong> specifies the character that is used to separate the field values.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>Operator</strong> specifies the operator to be used in the expression. The valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>EQUAL_TO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>NOT_EQUAL_TO</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>GREATER_THAN</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>GREATER_THAN_EQUAL</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>LESS_THAN</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>LESS_THAN_EQUAL</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>IN_LIST</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>REGEX</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>FieldName</strong> specifies the field name. If you specify <strong>CHARGING_RESULT_SPEC</strong> as the <strong>FieldKind</strong>, the valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>TAX_CODE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>GLID</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>RUM</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>BALANCE_ELEMENT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>ZONE_RESULT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>TIMEMODEL_TAG_NAME</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>GENERIC_SELECTOR_RESULT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>ORIG_ZONE_RESULT</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>PRICING_NAME</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>TIME_MODEL_NAME</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>ZONE_MODEL_NAME</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>EVALUATED_ZONE_MODEL_NAME</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>EVALUATED_ZONE</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>CHARGE_RATE_PLAN_NAME</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>FieldKind</strong> specifies the type of attributes to be selected for assigning the balance impacts. The valid values are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>PRODUCT_SPEC_FIELD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>EVENT_SPEC_FIELD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>CUSTOMER_SPEC_FIELD</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>CHARGING_RESULT_SPEC</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- <strong>FieldValue</strong> specifies the field value that is used for tracking balance impacts.</td>
</tr>
</tbody>
</table>
Loading the OOB_ItemTypeSelector.XML File

To load the OOB_ItemTypeSelector.xml file:

1. If you have created custom bill items in BRM, make a backup copy of the customized config_item_tags.xml and config_item_types.xml files in BRM.

2. Go to the PDC_home/apps/Samples/Examples directory.

3. Import the default billing item assignments from the OOB_ItemTypeSelector.xml file into the PDC database by running the following command:

```
ImportExportPricing -import -config /OOB_ItemTypeSelector.xml -ow
```

The item type selector is imported into the PDC database and the existing item tag-to-item type mapping in the BRM database is overwritten.

Configuring Item Type Selectors

To configure an item type selector:

---

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>itemSpec</td>
<td><code>&lt;itemSpec&gt;</code></td>
<td>Contains the details about the item specification, where:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;name&gt;</code>*ItemSpecificationName&lt;/name&gt;`</td>
<td>■ ItemSpecificationName specifies the item specification name. The item specification name is associated with the item tag field in BRM.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;description&gt;</code>*ItemSpecificationDescription&lt;/description&gt;`</td>
<td>■ ItemSpecificationDescription contains the description of the item specification.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;priceListName&gt;</code>*PriceListName`</td>
<td>■ PriceListName specifies the price list name. The valid value is Default.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;type&gt;</code>*ItemType`</td>
<td>■ ItemType specifies the type of the bill item for assigning balance impacts. This must match the item type created in BRM.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;default&gt;</code>*DefaultOption`</td>
<td>■ DefaultOption specifies whether a item specification is the default specification for the bill item. The valid values are true and false.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;aggregationType&gt;</code>*AggregationType`</td>
<td>■ AggregationType specifies whether the item accumulates charges or tracks each charge separately. The valid values are:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;category&gt;</code>*ChargeCategory`</td>
<td>■ Category specifies the charge category. See “Configuring Charges in Charge Offers” for more information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: You can add multiple item specifications, but you can add only one default item specification. You can use the same item specification in multiple item type selectors, but the item specification name must be unique.</td>
</tr>
</tbody>
</table>
1. Create the custom bill item in the database by subclassing the /item storable object. For example, you can create an /item/national object for tracking the charges for national calls. See the discussion about creating custom fields and storable classes in BRM Developer’s Guide.

2. Make a copy of the PDC_home/apps/Samples/Examples/SampleItemTypeSelector.xml file to customize.

3. Open the file in a text editor or XML editor.

4. Add or modify the elements as required. See Table 21–1 for the usage and description of each element.

5. Save and close the file.

---

**Note:** You can save the file with a different name and location or use the original file.

---

6. Go to the directory where you have saved the file.

7. Import the item type selector from the customized XML file into the PDC database by running the following command:

   ```bash
   ImportExportPricing -import -config filename
   ```

   where `filename` is the name of the customized XML file.

   The item type selector is imported into the PDC database and transformed to BRM and ECE.
This document describes how to configure dynamic quota in Pricing Design Center (PDC) to allocate the available quota dynamically for each parallel session of a subscriber based on the rules you configure.

Topics in this document:
- Configuring Dynamic Quota Selectors

**Configuring Dynamic Quota**

---

**Note:** You can use Dynamic Quota Configuration only in Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE).

Dynamic quota allows you to allocate the available quota dynamically for each parallel session of a subscriber based on the rules you configure. This enables the subscribers to run concurrent online charging sessions effortlessly. This also enables you to optimize the network usage effectively. For more information on dynamic quota, see *ECE Release Notes*.

You can define the rules for determining the dynamic quota allocation by configuring dynamic quota selectors in PDC. The rules can reference any of, or a combination of, the event, service, and customer attributes, and profile attribute specifications. You can configure the dynamic quota selector for each service and event combination.

ECE evaluates the rules defined in the dynamic quota selector at run time to derive the quota to be allocated for a session and the quota attributes, such as quota holding time (QHT), volume quota threshold (VQT), and quota validity time (VT). Dynamic quotas change in real time based on the dynamic quota selector rules as ECE grants and redistributes the quotas.

You configure dynamic quota selectors by importing the rules in XML files by using the `ImportExportPricing` utility. You can use the `SampleDynamicQuotaSelector.xml` file in the `PDC_home/apps/Samples/Examples` directory (where `PDC_home` is the directory in which the PDC software is installed) to create the XML files for configuring dynamic quota selectors. See "Configuring Dynamic Quota Selectors" for more information.

Table 22-1 lists the elements that are supported in the dynamic quota selector XML file, the usage of each element, and a description of how to specify each element in the XML file.
### Table 22–1 Elements Supported in the Dynamic Quota Selector XML File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td><code>&lt;name&gt;DynamicQuotaConfiguration&lt;/name&gt;</code></td>
<td>DynamicQuotaConfiguration specifies the dynamic quota selector name. For example, Generic Selector ABC.</td>
</tr>
<tr>
<td>description</td>
<td><code>&lt;description&gt;DynamicQuotaSelectorDescription&lt;/description&gt;</code></td>
<td>DynamicQuotaSelectorDescription specifies the description of the dynamic quota selector.</td>
</tr>
<tr>
<td>priceListName</td>
<td><code>&lt;priceListName&gt;PriceListName&lt;/priceListName&gt;</code></td>
<td>PriceListName specifies the price list name. The valid value is <code>Default</code>.</td>
</tr>
<tr>
<td>applicableToName</td>
<td><code>&lt;applicableToName&gt;ApplicableToName&lt;/applicableToName&gt;</code></td>
<td>ApplicableToName specifies the service name. The attributes of this service are used in the rules.</td>
</tr>
<tr>
<td>eventSpecName</td>
<td><code>&lt;eventSpecName&gt;EventName&lt;/eventSpecName&gt;</code></td>
<td>EventName specifies the name of an event. The attributes of this event are used in the rules.</td>
</tr>
<tr>
<td>validityPeriod</td>
<td><code>&lt;validFrom&gt;Validfromdate&lt;/validFrom&gt;</code></td>
<td>Validfromdate specifies the date from which the rule is valid. The valid format is <code>YYYYMMDD</code>. <strong>Note:</strong> It is mandatory to mention the date from when the rule is valid. If no date is specified, the rule is valid from immediate effect.</td>
</tr>
</tbody>
</table>
| applicableToAllChildEvent | `<applicableToAllChildEvent>ChildEventsOption</applicableToAllChildEvent>` | Specifies whether the dynamic quota selector is applicable to the child events, where ChildEventsOption is:  
  - **True** if the dynamic quota selector applies to all the child events.  
  - **False** if the dynamic quota selector applies to only the specific event type. |
| rule                   | `<rule>RuleName</rule>`                                               | Specifies the rules for allocating dynamic quotas, where:  
  - **RuleName** specifies the rule name.  
  - **RuleOrder** specifies the order in which the rules must be applied.  
  - **DynamicQuotaConfiguration** contains the configurations for dynamic quotas.  
  - **RequestedUnitsForAllocation** contains the requested units for allocating dynamic quotas.  
  - **FieldToValueExpression** contains the attributes used for deriving the quota.  
  - **ComplexExpression** contains the complex conditions for deriving the quota. **Note:** You can add multiple rules in the dynamic quota selector. Each rule in the dynamic quota selector can refer to a different attribute. For example, Rule 1 can use an event attribute, Rule 2 can use a balance expression, and Rule 3 can use a service attribute. |
### DynamicQuotaConfiguration

<configuration>
  <key>key</key>
  <value>value</value>
  <unit>unit</unit>
</configuration>

Specifies the quota attribute configurations, where:
- **key** specifies the quota attribute. The valid values are:
  - VOLUME_QUOTA_THRESHOLD.
  - QUOTA_HOLDING_TIME.
  - VALIDITY_TIME.

- **value** specifies the quota attribute value.
- **unit** specifies the unit for measurement. The valid values are:
  - Seconds
  - Minutes
  - Hours
  - Days
  - Bytes
  - Kbytes
  - Mbytes
  - Gbytes
  - None

Note: You must add the DynamicQuotaConfiguration element for each valid quota attribute.

### RequestedUnitsForAllocation

<requestedUnits>
  <fieldName>fieldname</fieldName>
  <unit>unit</unit>
  <dynamicQuotaBinaryExpression>dynamicQuotaBinaryExpression</dynamicQuotaBinaryExpression>
</requestedUnits>

<requestedUnits>
  <fieldName>fieldname</fieldName>
  <unit>unit</unit>
  <numberExpression>numberExpression</numberExpression>
</requestedUnits>

Specifies the details about the requested units, where:
- **fieldName** specifies the fully qualified requested attribute field name. This is specified in the event specification. For example, EventDelayedSessionTelcoGsm.REQUESTED_UNITS.INPUT_VOLUME.
- **unit** specifies the unit for measurement. The valid values are:
  - Seconds
  - Minutes
  - Hours
  - Days
  - Bytes
  - Kbytes
  - Mbytes
  - Gbytes
  - None

- **dynamicQuotaBinaryExpression** specifies the condition for deriving the quota.
- **numberExpression** specifies a valid number; for example, 10.0.
### Table 22–1 (Cont.) Elements Supported in the Dynamic Quota Selector XML File

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
</table>
| `dynamicQuotaBinaryExpression` | `<dynamicQuotaBinaryExpression>` | Specifies the condition for deriving the quota, where:  
- `numberExpression` specifies a valid number; for example, 10.0.  
- `balanceExpression` specifies the currency or noncurrency balance element defined in PDC.  
- `Operator` specifies the arithmetic operators. The valid values are: Add, Subtract, Multiply, Divide |
| `FieldToValueExpression` | `<dynamicQuotaFieldToValueExpression>` | Specifies the attributes used for deriving the quota, where:  
- `Delimiter` specifies the character that is used to separate the field values. **Note:** This is applicable only when the `operator` is `IN_LIST`.  
- `Operator` specifies the operator to be used in the expression. The valid values are: EQUAl_TO, N OT_EQUAL_TO, GREATER_THAN, GREATER_THAN_EQUAL, LESS_THAN, LESS_THAN_EQUAL, IN_LIST, CONTAINS  
- `FieldName` specifies the fully qualified field name. For example, `EventSessionTelcoGsm.REQUESTED_UNITS.NUMBER_OF_UNITS`.  
- `FieldKind` specifies the type of attributes. The valid values are: PRODUCT_SPEC_FIELD, EVENT_SPEC_FIELD, CUSTOMER_SPEC_FIELD, PROFILE_SPEC_FIELD  
- `FieldValue` specifies the field value that is used to track the balance impacts. **Note:** You can add multiple `FieldToValueExpression` elements in a rule to derive the quota based on a combination of attributes. |
Configuring Dynamic Quota Selectors

To configure a dynamic quota selector:

1. Make a copy of the `PDC_home/apps/Samples/Examples/Sample_DynamicQuotaSelector.xml` file to customize.
2. Open the file in a text editor or XML editor.
3. Add or modify the elements as required. See Table 22–1, "Elements Supported in the Dynamic Quota Selector XML File" for the usage and description of each element.
4. Save and close the file.

---

**Note:** You can save the file with a different name.

---

5. Go to the directory where you have saved the file.
6. Import the dynamic quota selector from the customized XML file into the PDC database by running the following command:

   `ImportExportPricing -import -config filename`

   where `filename` is the name of the customized XML file. For example, the `Dynamic_QuotaSelector.xml` file.

The dynamic quota selector is imported into the PDC database and published to ECE.

---

**Table 22–1 (Cont.) Elements Supported in the Dynamic Quota Selector XML File**

<table>
<thead>
<tr>
<th>Element</th>
<th>Syntax</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComplexExpression</td>
<td><code>&lt;dynamicQuotaComplexExpression&gt;</code></td>
<td>Specifies complex conditions for deriving the quota, where:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;operator&gt;Operator&lt;/operator&gt;</code></td>
<td>■ <code>Operator</code> specifies the operator to be used in the expression. The valid values are:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;value&gt;Value&lt;/value&gt;</code></td>
<td>■ <code>Value</code> is a numeric value. For example, 0.004.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;dynamicQuotaBinaryExpression&gt;</code></td>
<td>■ <code>dynamicQuotaBinaryExpression</code> specifies the condition for deriving the quota.</td>
</tr>
</tbody>
</table>

---

**ComplexExpression**

`<dynamicQuotaComplexExpression>`<br>
`<operator>Operator</operator>`<br>`<value>Value</value>`
`<dynamicQuotaBinaryExpression>`
`<dynamicQuotaBinaryExpression>`
`</dynamicQuotaComplexExpression>`
This document describes how to use folds in Oracle Communications Billing and Revenue Management (BRM) product offerings.

Topics in this document:

- About Folds
- About Applying Folds Only For Account-Level Products

See also:

- Managing Balances during Charging
- About Creating Product Offerings

About Folds

When a fold event occurs, BRM checks whether the balance is valid or expired. The balance is folded if the sub-balance is expired or if the sub-balance is always valid.

For example,

- 500 minutes are valid forever; therefore the balance can be folded at any time as shown in Figure 23–1.

![Figure 23–1 Minutes Valid Forever](image)

- 500 minutes that expire on February 1 can be folded after January 31 23:59:59. The end time of the fold event must be January 31 23:59:59 or later as shown in Figure 23–2.

![Figure 23–2 500 Minutes Expiring on February 1st](image)
About Applying Folds Only For Account-Level Products

**Important:** If you use delayed billing, folds occur at the end of the delayed billing period. This is to ensure that delayed events can continue to consume granted balances before final billing is run. However, free balances are granted at the beginning of each cycle, before the fold event occurs. If the newly granted balances do not have a configured validity period (that is, their validity period never ends), the fold event will remove those balances.

When you use fold events to remove unused balances and you also use delayed billing, you should always configure a validity period for the granted balances. This prevents fold events from removing balances that are granted at the beginning of a cycle before final billing is run.

---

**About Applying Folds Only For Account-Level Products**

If a fold is configured at account level, the fold is applicable to all the services in the account. The fold is then applied to the first service that is retrieved. If the resource balance is non-zero after applying the fold to the first service, the fold is applied again to the next service until the resource balance is zero.

To apply the fold only to account-level products, you can configure an account-level balance group to track the resource that you want to fold. The fold is then applied to only the products associated with the account-level balance group and not to all the services in the account.
This document describes how to implement and manage credit limits in Oracle Communications Billing and Revenue Management (BRM) product offerings.

Topics in this document:
- About Credit Limits
- About Credit Thresholds and Floors
- Handling Credit Limit Conflicts

See also:
- Managing Balances during Charging
- Managing Sub-Balances
- About Creating Product Offerings

About Credit Limits

A credit limit is the maximum amount of a balance element that can accumulate in a balance group. When a credit limit is reached, businesses typically deny customers access to the services associated with the balance group. For example, you might set a credit limit of $100 for a telephony package and deny service when customers who reach that limit try to place a call.

You set credit limits for balance elements in packages.

About Credit Thresholds and Floors

You can use credit thresholds to notify customers when they are approaching the credit limit of a balance. A credit threshold specifies the balance total that triggers a notification to the customer. You can specify the threshold in the following ways:

- As a fixed value, such as $100 or 30 minutes.
- As a percentage of the credit limit, such as 90%. For example, if the credit limit is $100 and the threshold is 90%, the threshold amount is reached when the customer has a balance of $90 (that is, when the customer has used 90% of the balance).

The credit floor is the starting point for a credit threshold and is the lowest number that the balance can be (that is, the number that represents no use of the balance).

For currency balances, the credit floor is 0.

For noncurrency balance elements, such as prepaid hours, you must specify a credit floor. You can use a negative number for the floor. For example, suppose you give 100
prepaid hours and set the credit limit to 0. When the credit limit is reached, the customer has no hours remaining and cannot use the service. To notify the customer when only 10 hours remain, set the credit threshold and floor as follows:

- Set the credit floor to -100. This number indicates none of the balance has been used.
- Set the credit threshold to 90%.

The threshold is reached at 90% of -100 hours (that is, when the customer has 10 prepaid hours left).

The credit threshold can be triggered both when a balance is increasing and when it is decreasing. You can customize BRM to perform different actions in each case. For example, if the credit threshold is crossed when the balance is increasing, service could be turned off. If the threshold is crossed when the balance is decreasing, service could be restored.

### Handling Credit Limit Conflicts

There might be occasions when customers purchase packages with credit limits that conflict with the credit limit set on their account or another service. There are two ways to handle credit limit conflicts:

- When you create your product offerings, if two services have different credit limits, create new balance groups for the services. This enables credit limits for each service to be set and tracked independently. See "Tracking Balances in Balance Groups".
- Specify which credit limit takes precedence when a new credit limit conflicts with an existing credit limit. You do this by setting the `credit_limit_conflict` entry in the CM configuration file.

To specify credit limit precedence:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Add the `credit_limit_conflict` entry:

   ```
   -fm_bill credit_limit_conflict value
   ```

   where value is one of these values:

   - `replace`, to use the new credit limit.
   - `ignore`, to ignore the new credit limit and keep the existing credit limit.
   - `add`, to add the new credit limit to the existing credit limit and create a new limit.
   - `minimum`, to use the credit limit that specifies the smaller amount.
   - `maximum`, to use the credit limit that specifies the greater amount.

   If this entry is not present, the new credit limit is used by default.

3. Stop and restart the CM.
This document describes how rollovers are managed in Oracle Communications Billing and Revenue Management (BRM) product offerings.

Topics in this document:

- About Rollovers
- When Rollover Events Occur
- About Rolling Over Noncurrency Balances during Package Transition
- About Rolling Over Balances That Expire in Midcycle
- Prorating Rollover Balances
- Rollover Example

See also:

- Managing Balances during Charging
- Managing Sub-Balances
- About Folds
- Configuring Balance Impact Rounding
- About Creating Product Offerings

**About Rollovers**

A rollover is a type of charge that credits unused balances at the end of a billing cycle to the account balance for the next cycle. For example, minutes are often rolled over from one month to the next. Balances from rollovers are different from granted balances, which are valid between two specific dates.

You define rollovers in charge offers in PDC. When defining rollovers, you specify the following:

- The period in which the rollover is valid. You can specify start and end time dates.
- The balance to roll over.
- The type of cycle to which the rollover applies (for example, monthly).
- The maximum number of cycles that a balance can be rolled over to.
- The maximum amount that can be rolled over in a cycle.
- The maximum value for a rollover balance.
- A G/L ID for the rollover charge.
BRM maintains each rollover balance as a sub-balance. BRM validates whether a rollover sub-balance can be rolled over by checking the rules that govern the rollover (such as the amount to roll over, the maximum rollover amount allowed, the maximum number of cycles to roll over, and so on).

When a rollover event occurs, one of three things happens to the rollover sub-balance:

- **If the full amount is eligible for rollover**, BRM does one of the following:
  - If the existing rollover sub-balance has the same properties as the new rollover sub-balance (such as the balance type, rollover rules, valid-from and valid-to dates, sub-balance contributors, and so on), BRM adds the rolled over amount to the existing rollover sub-balance.
  - If the existing rollover sub-balance has different properties then the new rollover sub-balance, BRM creates a new rollover sub-balance for the rolled over amount. The rollover sub-balance validity period has the same valid-from date as the original sub-balance, and its valid-to date is extended to the end of the new cycle.

- **If only a portion of the balance is eligible for rollover**, the amount in the rollover sub-balance is divided into a nonrollover sub-balance and a rollover sub-balance. The nonrollover sub-balance has the same valid-to date as the original rollover sub-balance. Its balance is used for late-arriving usage events. The valid-to date for the rollover sub-balance is extended to the end of the new cycle. Its balances are available for the customer to use in the new cycle.

- **If none of the balance is eligible for rollover**, the rollover sub-balance is not changed. This condition occurs when the rollover sub-balance has already been rolled over the maximum number of times allowed. Amounts in this rollover sub-balance are available only for late-arriving usage events.

---

**Important:** BRM memory limits might limit the number of months you can roll over a rollover sub-balance. The number of rollover sub-balances that BRM must maintain varies according to the number of rollover cycles allowed. You set the maximum number of rollover cycles allowed when configuring the rollover balance impact in PDC.

---

**When Rollover Events Occur**

Rollover sub-balances can be rolled over to another cycle as soon as they surpass the valid-to date (expire). Most rollover sub-balances expire on the account’s billing day of month (DOM). However, some rollover sub-balances may expire in the middle of a cycle due to a charge offer or service cancellation or due to flexible cycles.

Because many BRM features depend on rollover sub-balances being rolled over the day after they expire, BRM rolls over balances at the following times:

- **At the end of the billing cycle:** BRM automatically rolls over all eligible rollover sub-balances to the next cycle as part of the billing process. This catches all rollover sub-balances that expire on the account’s billing DOM.

- **When you run the pin_bill_day script:** The pin_bill_day script automatically executes the pin_rollover utility as part of the billing process. The pin_rollover utility rolls over any rollover sub-balances that have expired but have not been rolled over to the next cycle. This catches any rollover sub-balances that expired in the middle of the cycle.
About Rolling Over Noncurrency Balances during Package Transition

When your customers transition from one package to another, you can specify that noncurrency balances be rolled over from one package to another if both packages are in at least one package list together.

To specify that noncurrency balances rollover between packages during package transition, you must perform these tasks:

- If you use a custom application for customer management, provide the trigger for controlled rollover in the input flist to the PCM_OP_SUBSCRIPTION_TRANSITION_PLAN opcode.
- When you create your package lists, ensure that packages between which you want to allow rollovers are in at least one package list together.

These general rules apply to controlled rollovers:

- Controlled rollovers are not affected by the charge offer cycle rollover settings.
- A controlled rollover does not count as a rollover sub-balance and hence is not restricted to the maximum rollover quantity and the rollover units per period settings for the cycle rollover.
- The cycle grants of noncurrency balances that have been prorated during the package transition are rolled over to the next package.

About Rolling Over Balances That Expire in Midcycle

A balance can expire in the middle of a cycle: for example, when the balance is valid for only minutes, hours, or days or when the balance is valid for one or more months and starts in the middle of a month.

If a balance’s validity period does not end when the cycle ends, the rollover sub-balance is valid for the entire cycle in which the balance is granted and for the whole of the next cycle.

In Figure 25–1, a monthly cycle event grants 60 minutes that are valid for two weeks from the grant date. The minutes are granted on January 1. On January 14, when the balance of minutes expires, any remaining balances that can be rolled over are added to a new rollover sub-balance, which is valid from January 1 to March 1:

<table>
<thead>
<tr>
<th>1/1</th>
<th>2/1</th>
<th>3/1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grant balance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rollover balance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The same rollover period applies when the validity period of the minutes ends after the cycle in which they are granted. In Figure 25–2, a monthly cycle event grants 300 minutes that are valid for six week from the grant date. The minutes are granted on January 1. On February 11, when the balance expires, any remaining balances that can be rolled over are added to a new rollover sub-balance, which is valid from January 1 to March 1:

If the balance’s validity period does not start at the beginning of the last cycle, the balances are rolled over for one entire cycle.
Prorating Rollover Balances

Customers typically purchase and cancel charge offers at some point in the middle of a cycle. When you set up a rollover in PDC, you can specify whether rollover sub-balances are prorated for the first and last cycles based on the number of days in the cycles that the charge offer is owned.

Table 25–1 describes the rollover proration options:

<table>
<thead>
<tr>
<th>Proration Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll over entire amount</td>
<td>Roll over the available monthly rollover balances.</td>
</tr>
<tr>
<td>No rollover</td>
<td>Do not roll over any available monthly rollover balances.</td>
</tr>
<tr>
<td>Prorate rollover amount</td>
<td>Calculate the rollover balances based on the percentage of the cycle that the customer owned the rollover.</td>
</tr>
</tbody>
</table>

If you choose to prorate the rollover amount, BRM uses the equation in Figure 25–3 to calculate the amount to roll over:

\[
\text{ValidFrom} - \frac{\text{ValidTo}}{\text{DaysInCycle}} \times \text{Rollover}
\]

where:

- \(\text{ValidFrom}\) is the validity period’s starting date. For example, if the validity period is from March 15 to April 14, ValidFrom is March 15.
- \(\text{ValidTo}\) is the validity period’s ending date. For example, if the validity period is from March 15 to April 14, ValidTo is April 14.
- \(\text{DaysInCycle}\) is the number of days in the current cycle. For example, if the validity period is from March 15 to April 14, DaysInCycle is 31.
- \(\text{Rollover}\) is the available monthly rollover balance, such as 100 minutes.

For example, suppose:

- A customer buys a charge offer on January 15 that grants 500 minutes of use during each cycle.
- A rollover is set up so that up to 200 unused minutes roll over into the following cycle.
- The usage cycle is a calendar month.
The customer does not use any minutes during the first cycle.

Table 25–2 describes how the rollover is handled, depending on the purchase midcycle proration setting:

**Table 25–2  Impact of Proration Options**

<table>
<thead>
<tr>
<th>Proration Option</th>
<th>Amount Rolled Over into February</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roll over entire amount</td>
<td>200 minutes are rolled over from January into February.</td>
</tr>
<tr>
<td></td>
<td>In February, the customer has 700 minutes (200 rolled over plus the</td>
</tr>
<tr>
<td></td>
<td>500 minutes February grant).</td>
</tr>
<tr>
<td>No rollover</td>
<td>No minutes are rolled over from January into February.</td>
</tr>
<tr>
<td></td>
<td>In February, the customer has only the February grant of 500 minutes.</td>
</tr>
<tr>
<td>Prorate rollover amount</td>
<td>Approximately 110 minutes are rolled over from January into February:</td>
</tr>
<tr>
<td></td>
<td>(17/31) * 200 = 109.67</td>
</tr>
<tr>
<td></td>
<td>In February, the customer has about 610 minutes (110 rolled over</td>
</tr>
<tr>
<td></td>
<td>plus the 500 minutes February grant).</td>
</tr>
</tbody>
</table>

A similar set of results applies to the last-month proration calculation specified in the cancel midcycle proration setting.

**Rollover Example**

In this example, the customer’s charge offer includes:

- 500 minutes granted at the beginning of each usage cycle.

- A rollover specifying that:
  - Up to 100 unused minutes from each month can be rolled over.
  - The maximum number of rollover cycles is 2.
  - A limit of 150 total rollover minutes from previous months can be rolled over into a new month.

Other assumptions:

- Cycles are monthly and start on the first day of each calendar month.
- The charge offer with the rollover is valid starting January 1.
- The customer consumes no minutes until March.

1. On January 1, the customer is granted 500 minutes for use during January as shown in Figure 25–4. The minutes are maintained in a sub-balance.
2. On February 1:
   - The cycle forward event creates a new sub-balance to track the grant of 500 minutes for use during February as shown in Figure 25–5.

   ![Figure 25–5 February Minutes Grant](image)

   - The cycle rollover event creates a new sub-balance for tracking unused January minutes that roll over into February. 100 minutes are put in the new sub-balance, and they are valid until March 1. The original sub-balance for January is decremented 100 minutes, leaving 400 minutes available for late-arriving January events as shown in Figure 25–6.

   ![Figure 25–6 January Minutes Rollover](image)

   The customer now has 600 minutes (500 granted February 1 plus 100 rolled over from January) available for use during February as shown in Figure 25–7.
3. On March 1:
   - The cycle forward event creates a new sub-balance to track the grant of 500 minutes for use during March as shown in Figure 25–8.

   **Figure 25–8 March Minutes Grant**

   - The cycle rollover event creates a new sub-balance for tracking unused February minutes that roll over into March as shown in Figure 25–9. 100 minutes are put in the new sub-balance, and they are valid until April 1. The original sub-balance for February is decremented by 100 minutes, leaving 400 minutes available for late-arriving February events.
The cycle rollover event divides the January rollover minutes into two sub-balances to enforce the rule that only 150 total rollover minutes for a balance can carry forward into a new month. Because 100 free February minutes are already rolled over, only 50 minutes can be rolled over from the January rollover sub-balance for use in March.

One new sub-balance contains 50 minutes available for use for late-arriving February and January events. The other contains 50 rollover minutes that the customer can use in March. The balances for each month are shown in Figure 25–10.

The customer now has 650 minutes (500 granted March 1 plus 100 rolled over from February plus 50 rolled over from January) available for use during March as shown in Figure 25–11.
4. During March, the customer consumes 620 minutes as shown in Figure 25–12. Sub-balances are used starting with the newest sub-balance, as indicated by the sub-balance valid-from dates:

- All 500 minutes from the March grant are consumed, leaving a zero balance.
- All 100 February rollover minutes are consumed, leaving a zero balance.
- 20 of the 50 January rollover minutes are consumed, leaving 30.

5. On April 1, the cycle forward event creates a new sub-balance to track the grant of 500 minutes for use during April as shown in Figure 25–13.

No minutes roll over into April from the March and February rollovers because these balances were consumed in March. The 30 minutes remaining from the January rollover are not rolled over into April because the maximum number of cycles a grant can be rolled over is set to 2.
The customer has only 500 minutes available for April.

**Important:** If two charge offers contributing to the same balance group have rollover charges configured for the same balance with the same rollover frequency, either of the charge offers can be used to roll over the balances. In this case, rollover results may vary depending on the charge offer that is selected first.
Configuring Balance Impact Rounding

This document describes how to configure balance impact rounding rules in Oracle Communications Billing and Revenue Management (BRM) product offerings.

Topics in this document:

- About Balance Impact Rounding
- Configuring Balance Impact Rounding
- Configuring Balance Impact Rounding in ECE

See also:

- Managing Balances during Charging
- Managing Sub-Balances
- About Creating Product Offerings

About Balance Impact Rounding

BRM enables you to create various rounding rules for different balance elements, events, and processes such as rating and discounting. You create rounding rules for several reasons:

- To increase the accuracy of rating and discounting results.
- To process usage fees more efficiently. For example, an infinite number such as 5.333... is more easily processed when it is rounded.
- To round for various currencies that use a different number of digits to the right of the decimal (for example, 10.25 dollars and 10 yen).
- To comply with currency conversion rules.
- To bill customers an amount that they can actually pay.

About Rounding Rules

You can configure balance impact rounding based on the following rounding criteria:

- The rounding scale is the number of significant digits to the right of the decimal point. For example, a scale of 2 applied to 10.321111 rounds to 10.32.
- The rounding mode defines whether the number is rounded up, down, or not at all, based on the value of the digit following the last significant digit. See "About Rounding Modes".
About Balance Impact Rounding

- The process to which the rounding configuration applies. You can specify one of these processes: rating, discounting, taxation, and accounts receivable (A/R). A/R includes actions such as billing, payments, adjustments, cancellations, and G/L reporting.

  Specifying the process enables you to round differently based on the operation. For example, you can round up using six decimal places for rating and round down using two decimal places for billing.

- The event type to which the rounding configuration applies. This enables you to round differently for events that represent specific types of usage, cycle fees, discounts, and rollovers. For example, US dollars and purchase events (/event/purchase).

Table 26–1 show how a US dollars balance element and purchase event combination can be rounded various ways for various processes:

<table>
<thead>
<tr>
<th>For a US Dollars Balance Element</th>
<th>For This Process</th>
<th>Use This Rounding Scale</th>
<th>Use This Rounding Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event type = /event/billing/product/fee/purchase</td>
<td>Rating</td>
<td>6</td>
<td>Down</td>
</tr>
<tr>
<td>Event type = /event/billing/product/fee/purchase</td>
<td>Discounting</td>
<td>6</td>
<td>Up</td>
</tr>
<tr>
<td>Event type = /event/billing/product/fee/purchase</td>
<td>A/R</td>
<td>2</td>
<td>Nearest</td>
</tr>
<tr>
<td>Event type = /event/billing/product/fee/purchase</td>
<td>Taxation</td>
<td>2</td>
<td>Nearest</td>
</tr>
</tbody>
</table>

You can specify any combination of scale and mode for balance element, event, and process combinations.

After each process that performs rounding (rating, discounting, and so forth), the balance impact of the event contains the rounded amount.

How BRM Applies Rounding

Rating, discounting, and taxation produce balance impacts, which are rounded and applied to the customer’s account balance. The balance impacts are rounded to the scale and mode configured for the event, balance element, and process combination.

Balance impacts for an account are stored in bill items, which are associated with the customer’s bill. When you run billing, the item totals are rounded according to the A/R rounding rule. Item totals are rounded before being added to the bill so that the bill itself never needs rounding. Before billing, item totals are unrounded and their amounts reflect the scale configured for the associated events.

Figure 26–1 show the general process of balance impact rounding:
Assuming that rounding is configured for all events and all processes (rating, discounting, taxation, and A/R), rounding is performed in the following order:

1. **Purchase and cycle fees.** When a package is purchased, the purchase and cycle fees are rounded, if necessary, and a balance impact is generated with the rounded amount. Purchase and cycle fees typically only need rounding when a user purchases or cancels an offer in the middle of the billing cycle. In this case, cycle fees and noncurrency grant amounts are prorated, if necessary, and then rounded.

2. **Usage fees.** As customers use their services, the usage events are rated, the usage fees are rounded, and balance impacts are generated with the rounded amounts.

3. **Discount fees.** If a usage discount applies, the discount is calculated on the rounded usage fee, and then the discount is rounded and a discount balance impact is generated with the rounded amount. If there are multiple usage discounts, each discount is rounded in sequence.

4. **Taxes.** If tax should be applied to the event, the tax is calculated on the rounded fee, and then the tax is rounded and a balance impact is generated with the rounded amount.

5. **Billing discounts.** When you run billing, if a billing-time discount applies, it is applied to the total of the rounded usage items, and then the discount is rounded and a discount balance impact is generated with the rounded amount.

6. **Bills.** When the bill is generated, the total amount in each item is rounded, and then all item totals are summed and included in the bill.

7. **Rollover.** If there are any rollover amounts, the rollover is calculated on the total rounded usage, and then the rollover amount is rounded and applied to the next billing cycle. For example, if 100 minutes can be rolled over and the customer used
60.4 minutes, the unused amount of 39.6 minutes is rounded and then added to the account balance for the next cycle.

To round for rollover, you configure a rounding rule for the noncurrency balance element and cycle forward event combination. Rollovers are typically rounded up.

**About Rounding and A/R Actions**

When entering amounts for A/R actions such as adjustments and refunds, customer service representatives (CSRs) typically use a natural scale: that is, the scale commonly used in the marketplace for that balance element. For example, a person who purchases a book using US dollars cannot make change smaller than one cent (.01 dollars), making 2 the natural scale for US dollars. However, if a CSR reverses or adjusts an event before billing, the scale used is the one in the event’s balance impact. By default, this is the natural scale, unless you change this to use a scale other than natural.

**About Rounding Billed and Unbilled Items**

The balance impacts of events associated with items can have a high scale. Before billing, item totals reflect the scale of their associated events. During billing, item totals are rounded using the A/R rounding configuration so that customers’ bills display the natural scale for the balance element. However, the events associated with billed items still retain their pre-billing scale.

If any operation is performed on billed items: for example, event and item adjustments, BRM rounds the balance impact of the operation according to the A/R rounding rule to maintain G/L integrity. Because actions on billed items are rounded when they occur, only pending items are rounded when billing is run.

**About Rounding for Specific Event Types**

When you configure a rounding rule for a specific event type, such as cycle or session events, that rounding rule applies to those events only for the process specified. If you do not configure a rule for every process, all other processes for that event type use the default rounding rule. The default rounding rule specifies natural scale for all events and the rounding mode most commonly used for the process. A rule for all events is specified by using an asterisk (*) as the default event type.

For example, given the configuration in Table 26–2, during rating, session events are rounded down and have a scale of 6. During taxation, however, session events use the default rule of rounding to the nearest with a scale of 2:

<table>
<thead>
<tr>
<th>Balance Element</th>
<th>Event Type</th>
<th>Process</th>
<th>Rounding Scale</th>
<th>Rounding Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>US dollars</td>
<td>/event/session</td>
<td>Rating</td>
<td>6</td>
<td>Down</td>
</tr>
<tr>
<td>US dollars</td>
<td>* (all events)</td>
<td>Taxation</td>
<td>2</td>
<td>Nearest</td>
</tr>
</tbody>
</table>

**About Rounding Counter Balances for Discounting**

For discount rounding, the rounding configurations are used to round only the balance impacts of discounting, not input balances, such as counters. To configure counter rounding, see “Configuring Rounding Rules for Counter Balances”.

26-4  PDC Creating Product Offerings
About G/L Report Rounding

G/L report rounding uses the rounding rule configured for A/R actions. The rounded totals in G/L reports might differ slightly from the total of the rounded bills. This is because item totals are rounded for billing, and journal entries are rounded for G/L reports. Also, G/L reports are rounded differently before and after billing. For more information, see "About Rounding and G/L Reports" in BRM Collecting General Ledger Data.

If there is any difference between the rounded journal entries in G/L reports and the rounded bill items, BRM records the difference in the G/L. You configure a G/L ID for the rounding difference and configure BRM to record the differences. See "Configuring to Record Rounding Differences in the G/L".

About Rounding Modes

A rounding mode defines whether a number is rounded up, down, or not at all. The rounding mode rounds to the specified scale. For example, using a scale of 2, rounding up 10.2369 results in 10.24. If the scale is 3, rounding up results in 10.237.

The following rounding mode values that you can use are defined in the BRM_home/include/pin_bill.h file:

- **0**: PIN_BEID_ROUND_NEAREST
  This mode rounds up or down depending on the value of the digit following the last significant digit. If the additional digit is 0-4, the last significant digit remains the same. If the additional digit is 5-9, the last significant digit is rounded up. For example, if the scale is 2, 10.144 rounds to 10.14 and 10.145 rounds to 10.15. This is the most common rounding method.

- **1**: PIN_BEID_ROUND_UP
  This mode rounds up when the digit following the last significant digit is greater than 0. For example, If the scale is 2, 10.151 rounds to 10.16. If the scale is 1, it rounds to 10.2.

- **2**: PIN_BEID_ROUND_DOWN
  This mode truncates all digits following the last significant digit. For example, if the scale is 2, 10.159 rounds to 10.15. If the scale is 1, it rounds to 10.1.

- **3**: PIN_BEID_ROUND_EVEN
  This mode rounds one of three ways depending on the value of the digit following the last significant digit:
  - If it is less than 5, truncate all digits following the last significant digit.
  - If it is greater than 5, round up.
  - If it is 5, round to the nearest even digit. For example, if the scale is 2, 10.155 and 10.165 both round to 10.16 because 6 is an even number.

- **4**: PIN_BEID_ROUND_FLOOR
  This mode rounds numbers toward a negative value (that is, the rounded number is always less than the unrounded number). This enables you to round balance impacts so that customers always benefit. For example, if the scale is 2, a credit to a customer of -7.999 is rounded to -8.00, and a debit of 7.999 is rounded to 7.99.

The following two modes perform the same rounding as their nonalternative counterparts (ROUND_FLOOR and ROUND_DOWN), except that they compensate for possible loss of precision when rounding down by first rounding...
with a mode of NEAREST using a scale that is two digits greater than the scale you configure.

- 5: PIN_BEID_ROUND_FLOOR_ALT
- 6: PIN_BEID_ROUND_DOWN_ALT

For more information, see "About Rounding Modes That Correct for Loss of Precision".

About Rounding Modes That Correct for Loss of Precision

Some calculations produce results that are slightly less than expected when a value is rounded down. For example, when BRM prorates a $60.00 cycle fee for 20 out of 30 active days, the calculation is \((20/30) \times 60.00\). The expected result is a fee of $40.00. However, because \(20/30\) evaluates to 0.666, when this is multiplied by 60 and rounded down, the actual result is a fee of $39.99.

BRM provides two alternative rounding modes that compensate for possible precision loss when rounding down: ROUND_DOWN_ALT and ROUND_FLOOR_ALT. These modes perform the same rounding as their nonalternative counterparts (ROUND_DOWN and ROUND_FLOOR) after first compensating for loss of precision.

When these modes are used, if a decimal should be rounded down, BRM performs two rounding functions: The decimal is rounded by using the ROUND_NEAREST rounding mode and a scale that is two more than the scale that you request. It is then rounded down.

For example, if you configure the rounding mode as ROUND_DOWN_ALT and a rounding scale of 2, and the decimal to round is 7.999..., BRM truncates the infinite decimal to the system maximum and then rounds this decimal to the nearest using a scale of 4 (2 more than the configured scale of 2), which results in 8.0000. This decimal is then rounded down using the configured scale of 2, resulting in 8.00 as shown in Table 26-3:

<table>
<thead>
<tr>
<th>Do This</th>
<th>Why</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.999...</td>
<td>Original decimal</td>
</tr>
<tr>
<td>7,9999999999999999</td>
<td>Truncated to the system max</td>
</tr>
<tr>
<td>8.0000</td>
<td>ROUND_NEAREST, using requested scale + 2</td>
</tr>
<tr>
<td>8.00</td>
<td>ROUND_DOWN to the requested scale of 2</td>
</tr>
</tbody>
</table>

Had the original decimal of 7.999... not been rounded to the nearest first and only rounded down, the result would be 7.99.

To compensate for possible loss of precision, the alternative rounding modes consider two decimal places more than the nonalternative rounding modes. Therefore, the greatest amount that will be modified by using the alternative rounding modes, and still compensate for loss of precision, is less than the greatest amount that will be modified by using the nonalternative rounding modes.
When Rounding Is Not Applied

When the requested rounding scale is greater than the scale of the number being processed, rounding is not required. This occurs when:

- You configure a rounding scale equal to or greater than the maximum number of digits allowed by the system. For example, if the system allows 15 digits and you set the rounding scale to 15 or greater, rounding has no effect.
- You call a rounding function and request a scale that is equal to or greater than the current scale of the decimal: for example, when the decimal is 10.89766 and the scale requested is 5 or greater.
- A computation or expression results in a decimal with a scale that is equal to or less than the configured or requested scale: for example, when a computation results in the decimal 1.98 and the configured scale is 2 or greater.

 Configuring Balance Impact Rounding

To set up rounding, perform the following tasks:

- Configuring Rounding Rules for Counter Balances
- Configuring to Record Rounding Differences in the G/L

About Configuring Rounding Rules

You configure rounding for balance element and event type combinations (for example, US dollars and /event/session/telco/gsm events). For each balance element and event combination, you specify a rounding scale, a rounding mode, and the process for which this rule applies.

---

**Important:** You do not need to configure rounding for all of the processes. However, you must configure rounding for every balance element and event type that has a balance impact.

---

About Rounding Mode Values in the BRM API

The rounding modes you specify have corresponding modes in the BRM API. When BRM invokes a decimal data type function, it converts the rounding mode in the /config/beid object into the corresponding rounding parameter used by the decimal data type function. If you call decimal data type functions in custom applications, specify the function’s rounding mode, not the BEID rounding mode.

Table 26–4 shows the BEID rounding modes and the corresponding rounding parameters for the decimal data type functions:

<table>
<thead>
<tr>
<th>BEID Rounding Mode (specified in pin_bill.h)</th>
<th>Rounding Mode Parameter for pbo_decimal functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>0: PIN_BEID_ROUND_NEAREST</td>
<td>5: ROUND_HALF_UP</td>
</tr>
<tr>
<td>1: PIN_BEID_ROUND_UP</td>
<td>1: ROUND_UP</td>
</tr>
<tr>
<td>2: PIN_BEID_ROUND_DOWN</td>
<td>2: ROUND_DOWN</td>
</tr>
<tr>
<td>3: PIN_BEID_ROUND_EVEN</td>
<td>7: ROUND_HALF_EVEN</td>
</tr>
<tr>
<td>4: PIN_BEID_ROUND_FLOOR</td>
<td>4: ROUND_FLOOR</td>
</tr>
</tbody>
</table>
Configuring Balance Impact Rounding

You use PDC to configure rounding rules for counter balances. When you configure discounts, use discount expressions to specify how to round the balance. Use the following discount expression syntax:

```
round(expression, rounding_scale, rounding_mode)
```

where `expression` defines the balance to round. This can be any discount expression. To round a counter balance, use the `Balance` expression. For more information, see the discussion about configuring discounts in PDC Help.

For example, to round a counter balance down to two decimal places for balance element ID 100099, use the following expression:

```
round(Balance(1000099), 2, ROUND_DOWN)
```

About Rounding Modes for Discount Expressions

Rounding modes for discount expressions have slightly different names. Table 26–5 lists the rounding mode values you can specify.

<table>
<thead>
<tr>
<th>Discount Expression Rounding Mode</th>
<th>Rounding Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROUND_PLAIN</td>
<td>Nearest</td>
</tr>
<tr>
<td>ROUND_UP</td>
<td>Up</td>
</tr>
<tr>
<td>ROUND_DOWN</td>
<td>Down</td>
</tr>
<tr>
<td>ROUND_BANKERS</td>
<td>Even</td>
</tr>
</tbody>
</table>

For a definition of what these modes represent, see "About Rounding Modes".

Configuring to Record Rounding Differences in the G/L

To record any difference between rounded bill items and the rounded total in the G/L, perform the following tasks:

- Defining a G/L ID for Rounding Differences
- Mapping the Rounding G/L ID to an Event
- Configuring BRM to Record Rounding Differences
For information about how rounding is performed in G/L reports, see "About Rounding and G/L Reports" in BRM Collecting General Ledger Data.

**Defining a G/L ID for Rounding Differences**

You define a G/L ID for rounding to include the rounding difference in G/L reports so that they can be accurately reconciled.

To define G/L IDs, you edit the G/L ID configuration file and then run the `load_pin_glid` utility to load the contents of the file into the `/config/glid` object in the BRM database.

---

**Important:** The `load_pin_glid` utility needs a configuration (pin.conf) file in the directory from which you run the utility.

---

To define a rounding G/L ID:

1. If necessary, edit the G/L ID configuration file in `BRM_home/sys/data/pricing/example/pin_glid`. If the following entry is not present, add it:

   ```
   #=================================================================
   # G/L ID for rounding adjustments
   #=================================================================
   glid
   id 1512
descr Rounding Epsilon
   gl_acct billed gross rounding.debit rounding.credit
   gl_acct billed net rounding.debit rounding.credit
   gl_acct billed disc rounding.credit rounding.debit
   gl_acct billed_earned gross rounding.debit rounding.credit
   gl_acct billed_earned net rounding.debit rounding.credit
   gl_acct billed_earned disc rounding.credit rounding.debit
   gl_acct unbilled gross rounding.debit rounding.credit
   gl_acct unbilled net rounding.debit rounding.credit
   gl_acct unbilled disc rounding.credit rounding.debit
   gl_acct unbilled_earned gross rounding.debit rounding.credit
   gl_acct unbilled_earned net rounding.debit rounding.credit
   gl_acct unbilled_earned disc rounding.credit rounding.debit
   gl_acct unbilled_earned disc rounding.credit rounding.debit
   ```

   **Caution:** The `load_pin_glid` utility overwrites existing G/L IDs. If you are updating G/L IDs, you cannot load new G/L IDs only. You must load complete sets of G/L IDs each time you run the `load_pin_glid` utility.

2. Save and close the file.

3. Use the following command to run the `load_pin_glid` utility:

   ```
   load_pin_glid pin_glid_file
   ```

   For more information, see "Loading General Ledger Configuration Data" in BRM Collecting General Ledger Data.

**Mapping the Rounding G/L ID to an Event**

Because the rounding difference is not a rated event, you must map the G/L ID to an event type. G/L ID mapping is defined in the `reasons.locale` file. You can find a sample
of this file in the `BRM_home/sysmsgs/reasoncodes` directory. The sample file is named `reasons.en_US` and contains the following default entry for the rounding G/L ID mapping:

```
DOMAIN = "Others" ;
STR
  EVENT-GLID
    ...
      /event/journal/epsilon*    1512 ;
  EVENT-GLID-END
```

**Note:** `/event/journal/epsilon` is a dummy event type used for reference only.

To change the G/L ID for rounding, you must edit and reload the file. The G/L ID you define in the `reasons.locale` and `pin_glid` files must match. See "Configuring to Record Rounding Differences in the G/L".

To map the G/L ID for rounding to an event, you use the `load_localized_strings` utility to load the contents of the file into the `/config/map_glid` object. When you run the `load_localized_strings` utility, use this command:

```
load_localized_strings reasons.locale
```

**Note:** If you are loading a localized version of this file, use the correct file extension for your locale.

**Caution:** The `load_localized_strings` utility overwrites the existing G/L ID maps. If you are updating this object, you cannot load new G/L ID maps only. You must load complete sets of G/L ID maps each time you run the `load_localized_strings` utility.

For information on loading the `reasons.locale` file, see "Creating a Localized Version of BRM" in *BRM Developer’s Guide*.

**Configuring BRM to Record Rounding Differences**

By default, rounding differences are not recorded in G/L reports. You can enable recording the difference between rounded bill items and the rounded total by running the `pin_bus_params` utility to change the `GenerateJournalEpsilon` business parameter. For information about this utility, see *BRM Developer’s Guide*.

To enable BRM to record rounding differences:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
3. In the file, change `disabled` to `enabled`:
   ```
   <GenerateJournalEpsilon>enabled</GenerateJournalEpsilon>
   ```
4. Save the file as `bus_params_billing.xml`. 
5. Load the XML file into the BRM database:
   
   ```
   pin_bus_params bus_params_billing.xml
   ```

6. Stop and restart the CM.

7. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see BRM System Administrator’s Guide.

### Rounding Examples

Rounding is performed after rating, discounting, taxation, and A/R actions such as billing and adjustments.

Table 26–6 shows the resulting balance impacts of rounded charges for rating, discounting, taxation, and billing. This example rounds to the nearest mode and uses these scales:

- 2 for purchase events, taxation, and A/R
- 5 for rating and discounting

#### Table 26–6  Rounding Examples

<table>
<thead>
<tr>
<th>Action / Process</th>
<th>Calculated Charge</th>
<th>Rounded Charge &amp; Balance Impact</th>
<th>Account Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purchase package</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate cycle fee</td>
<td>9.95</td>
<td>9.95</td>
<td>9.95</td>
</tr>
<tr>
<td>Use service</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rate usage</td>
<td>5.23456789</td>
<td>5.23457</td>
<td>15.18456</td>
</tr>
<tr>
<td>Discount usage fee</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10%</td>
<td>0.523456789</td>
<td>0.52346</td>
<td>14.66111</td>
</tr>
<tr>
<td>Tax usage event 3%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(after discount)</td>
<td>0.1413333</td>
<td>0.14</td>
<td>14.80111</td>
</tr>
<tr>
<td></td>
<td>= 3% * (rounded usage fee - rounded discount)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run billing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apply billing-time discount of 5% off total usage</td>
<td>0.24250...</td>
<td>0.2425</td>
<td>14.55861</td>
</tr>
<tr>
<td></td>
<td>= 5% * usage item total after A/R rounding</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(The usage total does not include the cycle fee.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create bill</td>
<td>14.55861</td>
<td>14.56</td>
<td>14.56</td>
</tr>
<tr>
<td></td>
<td>= Total of all items</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(No balance impact)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Correcting for Precision Loss When Rounding Down

This example shows the results of rounding when you use the ROUND_DOWN_ALT and ROUND_FLOOR_ALT modes. For more information, see "About Rounding Modes That Correct for Loss of Precision".

The ROUND_DOWN_ALT and ROUND_FLOOR_ALT modes produce different results than ROUND_DOWN and ROUND_FLOOR only when the three digits following the last significant digit are 995 or greater. (The last significant digit is the digit in the decimal place corresponding to the scale: If the scale is 2, the last significant digit in the number 1.23456 is 3.)
For example:

Table 26–7 shows some rounding results of the ROUND_DOWN_ALT and ROUND_FLOOR_ALT modes as compared to their nonalternative rounding counterparts (ROUND_DOWN and ROUND_FLOOR) for various decimal values and rounding scales.

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Scale</th>
<th>Rounding Mode</th>
<th>Rounding Mode</th>
<th>Rounding Mode</th>
<th>Rounding Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>NA</td>
<td>NA</td>
<td>DOWN</td>
<td>ROUND_DOWN</td>
<td>ROUND_FLOOR</td>
<td>ROUND_FLOOR_ALT</td>
</tr>
<tr>
<td>1.5256</td>
<td>2</td>
<td>1.52</td>
<td>1.52</td>
<td>1.52</td>
<td>1.52</td>
</tr>
<tr>
<td>-1.5256</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-2</td>
<td>-2</td>
</tr>
<tr>
<td>12.8999...</td>
<td>0</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>12.8999...</td>
<td>1</td>
<td>12.8</td>
<td>12.9</td>
<td>12.8</td>
<td>12.9</td>
</tr>
<tr>
<td>12.8999...</td>
<td>2</td>
<td>12.89</td>
<td>12.90</td>
<td>12.89</td>
<td>12.90</td>
</tr>
<tr>
<td>-12.8999...</td>
<td>1</td>
<td>-12.8</td>
<td>-12.9</td>
<td>-12.8</td>
<td>-12.9</td>
</tr>
<tr>
<td>-12.8999...</td>
<td>2</td>
<td>-12.89</td>
<td>-12.90</td>
<td>-12.89</td>
<td>-12.90</td>
</tr>
<tr>
<td>-6.9990</td>
<td>2</td>
<td>-6.99</td>
<td>-6.99</td>
<td>-7.00</td>
<td>-7.00</td>
</tr>
<tr>
<td>-6.9990</td>
<td>3</td>
<td>-6.999</td>
<td>-6.999</td>
<td>-6.999</td>
<td>-6.999</td>
</tr>
<tr>
<td>7.999...</td>
<td>0</td>
<td>7</td>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>7.999...</td>
<td>1</td>
<td>7.9</td>
<td>8.0</td>
<td>7.9</td>
<td>8.0</td>
</tr>
<tr>
<td>7.999...</td>
<td>2</td>
<td>7.99</td>
<td>8.00</td>
<td>7.99</td>
<td>8.00</td>
</tr>
<tr>
<td>-7.999...</td>
<td>0</td>
<td>-7</td>
<td>-8</td>
<td>-8</td>
<td>-8</td>
</tr>
<tr>
<td>-7.999...</td>
<td>2</td>
<td>-7.99</td>
<td>-8.00</td>
<td>-8.00</td>
<td>-8.00</td>
</tr>
</tbody>
</table>

**Table 26–7  Rounding Results**

Rounding Using Different Modes

The aggregated effects of rounding on the final balance impact is determined by the mode and scale that you configure. The higher the scale, the less effect the rounding mode has on the final balance impact.

For example, Table 26–8 shows the impact of various rounding mode combinations for rating a usage fee of $1.1234567 that includes a 10% discount. Both rating and discounting use a scale of 6:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Round for rating</td>
<td>1.123456</td>
<td>1.123456</td>
<td>1.123457</td>
<td>1.123457</td>
</tr>
<tr>
<td>Calculate 10% discount</td>
<td>.1123456</td>
<td>.1123456</td>
<td>.1123457</td>
<td>.1123457</td>
</tr>
<tr>
<td>Round discounted amount</td>
<td>.112345</td>
<td>.112346</td>
<td>.112345</td>
<td>.112346</td>
</tr>
</tbody>
</table>
In this example, the difference in the final balance impacts is small because the scale is high and probably will not change the final amount on the bill. However, when many events are summed in an item, or the scale is small, such as 2 or 3, the differences become greater.

**Tip:** If you calculate the discount without rounding the event, the configuration where rating is rounded down and discounting is rounded up returns the most accurate result. Therefore, this is the best mode configuration to use when you discount events.

### Configuring Balance Impact Rounding in ECE

You can configure rounding for a currency and noncurrency balance impact by specifying rules for how ECE rounds the rating impact amounts it calculates. The rules you specify are applied at the system level so the rounding of the rating impact amounts apply across all charges, discounts, and chargershares. The rounding is also applied to reverse rating calculations and taxation calculations.

The rules you can specify for rounding are as follows:

- **Scale**
  
  You can specify a rule for how many digits to the right of the decimal point to allow. The default scale is 2.

- **Rounding mode**
  
  You can specify a rule for the rounding behavior according to the Java math rounding enum. The default is **HALF_UP**.

  Go to the Java SE Technical Documentation Web site for information about using the Java math rounding enum:

  [http://docs.oracle.com/javase/6/docs/api/java/math/RoundingMode.html](http://docs.oracle.com/javase/6/docs/api/java/math/RoundingMode.html)

For information about configuring rounding for currency and noncurrency balance impacts, see "Configuring Rounding for a Currency Balance Impact" and "Configuring Rounding for a Noncurrency Balance Impact".

### Example of Currency Rounding for a Charge

If you allow two digits to the right of the decimal point and you round down towards zero (**DOWN** rounding mode), ECE takes a calculated charge of 0.509 USD and rounds it to 0.50 USDs.
Example of Noncurrency Rounding for a Charge
If you allow zero digits to the right of the decimal point and you round towards positive infinity (UP rounding mode), ECE takes a charge of 0.509 bonus point and rounds the value to 1 bonus point.

Examples of Currency Rounding for Discounts
If you allow zero digits to the right of the decimal point and you round down towards zero (DOWN rounding mode), ECE takes a discount of -2.5 USD and rounds the value to -2 USD.

If you allow zero digits to the right of the decimal point and you round towards negative infinity (FLOOR rounding mode), ECE takes a discount of -2.5 USD and rounds the value to -3 USD.

If you allow two digits to the right of the decimal point and you round down towards zero (DOWN rounding mode), ECE takes a discount of -0.075 USD and rounds the value to -0.07 USD.

Configuring Rounding for a Currency Balance Impact
You can configure rounding for a currency balance impact by specifying rules for how ECE rounds the rating impact amounts it calculates.

The rules you specify are applied as a systemwide configuration for rounding the rating impact amounts of all charges, discounts and chargeshares. The rounding is also applied to all reverse rating calculations and taxation calculations.

See "Configuring Rounding for a Noncurrency Balance Impact" for information about configuring rounding for noncurrency balance impacts.

To configure rounding for a currency balance impact:

1. Access the ECE configuration MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Connect to the ECE charging server node enabled for JMX management.
      This is the charging server node set to start CohMgt = true in the ECE_home/oceceserver/config/ecdTopology.conf file.
   d. Start a JMX editor that enables you to edit MBean attributes, such as JConsole.
   e. In the editor’s MBean hierarchy, find the ECE configuration MBeans.

2. Expand the ECE Configuration node.
3. Expand charging.server.
4. Expand Attributes.
5. Specify values for the following attributes:
   - currencyScale: Enter the number of digits you allow to the right of the decimal point for a calculated impact amount.
     For example, enter 2 if you allow two digits to the right of the decimal point.
     The default is 2.
   - currencyRoundingMode: Enter the rounding mode that determines the rounding behavior by entering the string representation of the Java math rounding enum.
Configuring Balance Impact Rounding in ECE

For more information, see the Java SE Technical Documentation Web site:

http://docs.oracle.com/javase/6/docs/api/java/math/RoundingMode.htm1

For example, enter **UP** to round up away from zero or **DOWN** to round down towards zero.

The default is **HALF_UP**.

6. Save your changes.

**Configuring Rounding for a Noncurrency Balance Impact**

You can configure rounding for a noncurrency balance impact. The rules you specify are applied as a systemwide setting and apply for rounding the rating impact amounts of all charges, discounts, and chargeshares. The rounding is also applied to all reverse rating calculations and taxation calculations.

To configure rounding for a noncurrency balance impact:

1. Access the ECE configuration MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Connect to the ECE charging server node enabled for JMX management.
      This is the charging server node set to **start CohMgt = true** in the ECE_home/oceceserver/config/eeetopology.conf file.
   d. Start a JMX editor that enables you to edit MBean attributes, such as JConsole.
   e. In the editor's MBean hierarchy, find the ECE configuration MBeans.

2. Expand the **ECE Configuration** node.
3. Expand **charging.server**.
4. Expand **Attributes**.
5. Specify values for the following attributes:
   - **nonCurrencyScale**: Enter the number of digits you allow to the right of the decimal point for a calculated impact amount.
     For example, enter 2 if you allow two digits to the right of the decimal point.
     The default is 2.
   - **nonCurrencyRoundingMode**: Enter the rounding mode that determines the rounding behavior by entering the string representation of the Java math rounding enum.
     For more information, see the Java SE Technical Documentation Web site:
     http://docs.oracle.com/javase/6/docs/api/java/math/RoundingMode.htm1
     For example, enter **UP** to round up away from zero or **DOWN** to round down towards zero.
     The default is **HALF_UP**.
6. Save your changes.
For information about configuring rounding for currency balance impacts, see "Configuring Rounding for a Currency Balance Impact”.

**Configuring Rounding When Authorizing Multiple RUMs**

When ECE receives an online charging request for a specified amount of usage, ECE uses the customers product offerings to determine if the customer’s balance amount is sufficient for the requested amount. If the request applies to multiple RUMs; for example, occurrence and duration, fractional values of the authorized balance quantity might result. You can enable ECE to round up the authorized balance quantities to the nearest whole number.

If your business requires that your customers must be able to use all of their balances, configure ECE to round up the authorized balance quantity. However, rounding up the authorized balance quantity may result in customers exceeding their credit limits.

You configure whether to round up the fractional value of the impacted balance quantity as a systemwide setting. ECE authorizes an additional RUM unit when the quantity is a fraction.

To configure rounding when authorizing multiple RUMs:

1. Access the ECE configuration MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Connect to the ECE charging server node enabled for JMX management.
      This is the charging server node set to `start CohMgt = true` in the `ECE_home/occeserver/config/eceTopology.conf` file.
   d. Start a JMX editor that enables you to edit MBean attributes, such as JConsole.
   e. In the editor’s MBean hierarchy, find the ECE configuration MBeans.
2. Expand the **ECE Configuration** node.
3. Expand **charging.server**.
4. Expand **Attributes**.
5. Set the **reverseRateUseAllBalances** attribute to one of the following values:
   - To round up the fractional value of the authorized balance quantity, enter **true**.
     This option allows customers to use all balances even if they might exceed their credit limits by a small amount.
   - To disallow the fractional value of the authorized balance quantity to be rounded up, enter **false**.
     This option does not allow customers to exceed their credit limits.
     The default is **false**.
6. Save your changes.
This part describes how to customize and maintain Pricing Design Center (PDC) and Oracle Communications Billing and Revenue Management (BRM) product offerings.

Part IV contains the following chapters:

- Enabling Charging for Custom Events
- Using the PDC Web Service
- Working with Profiles
- Importing and Exporting Pricing and Setup Components
- Replicating Data Between PDC Systems
- Pricing Utilities
Enabling Charging for Custom Events

This document describes how to enrich custom events in Pricing Design Center (PDC) with information required by Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) to process usage charging requests from the network.

Topics in this document:

- Overview of Enriching Event Definitions
- About Charging Operation Types
- About Charging Operation Type Versions
- About Mapping Network Attributes to Event Attributes
- Before Enriching Event Definitions
- Enriching Event Definitions
- Editing Event Definitions

See also:

- Configuring Charge Offers
- About Creating Product Offerings

Overview of Enriching Event Definitions

BRM includes a set of commonly-used event definitions, such as the events used by GSM and GPRS services. If you create custom events and services by using BRM Developer Center, you need to use PDC to enrich the event definitions with additional information required by ECE.

You do the following to enrich an event definition:

- Add USER_IDENTITY and CALLED_ID attributes. The USER_IDENTITY attribute is required for all events. The CALLED_ID attribute is required only if you use zoning, or other charging scenarios that are based on the call destination.

- Add transient attributes if required. These are attributes that are used specifically for usage charging and do not need to be stored persistently in the event for rerating. For example, you can add the CELL_ID attribute to retrieve cell load information, but the CELL_ID value does not need to be stored in the event object.

- Add usage charging operation types (for example, initiating a session-based charge for an event). See "About Charging Operation Types" for more information.
Define the network mapping for the events; for example, if you are using Diameter Gateway, you map the request attributes that are received from the Diameter network to event attributes. This mapping is used by the ECE charging client at runtime to populate the event. See "About Mapping Network Attributes to Event Attributes" for more information.

### About Charging Operation Types

A charging operation type is a type of operation that ECE uses for usage charging; for example, the Initiate operation type is used for initiating a session-based charge and the Price_Enquiry operation type is used for enquiring about price information. Each charging operation type requires a set of event attributes to be specified.

When you add the charging operation type in the event definition, you associate the charging operation type with the group of event attributes that are necessary to perform the charging operation on the event. For example:

- The attributes for session-based charging are associated with the Initiate, Update, and Terminate operation types.
- The attributes for event-based charging are associated with the Debit, Refund, and Price_Enquiry charging operation types.

Table 27–1 lists the charging operation types that are used in ECE.

<table>
<thead>
<tr>
<th>Operation Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate</td>
<td>Initiates a session-based charging operation.</td>
</tr>
<tr>
<td>Update</td>
<td>Modifies a session-based charging operation.</td>
</tr>
<tr>
<td>Terminate</td>
<td>Terminates a single non-session-based charging operation.</td>
</tr>
<tr>
<td>Cancel</td>
<td>Cancels a session-based charging operation.</td>
</tr>
<tr>
<td>Refund_Amount</td>
<td>Refunds a specific amount to a specific balance.</td>
</tr>
<tr>
<td>Refund_Unit</td>
<td>Refunds a calculated amount, based on units consumed, to the impacted balances.</td>
</tr>
<tr>
<td>Debit_Amount</td>
<td>Debits a specific amount to a specific balance.</td>
</tr>
<tr>
<td>Debit_Unit</td>
<td>Debits a calculated amount, based on Units consumed, to the impacted balances.</td>
</tr>
<tr>
<td>Price_Enquiry</td>
<td>Generates a price estimation without any balance reservations occurring. It is used when there isn’t a high probability of receiving a charging request. For example, Price_Enquiry might be called to get the price of an event charge to display in a content portal.</td>
</tr>
<tr>
<td>Start_Accounting</td>
<td>Starts tracking usage without incurring balance impacts.</td>
</tr>
<tr>
<td>Update_Accounting</td>
<td>Continues tracking usage without incurring balance impacts.</td>
</tr>
</tbody>
</table>

See "Enriching Event Definitions" for information on how to add charging operation types and associating them with the event attributes.

### About Charging Operation Type Versions

A set of charging operation types has a version, which is indicated by the <externalVersion> element.
When you modify the charging operation types in PDC, use the same version number to replace the existing set of charging operation types with the updated set of charging operation types. The updated charging operation types are published to ECE.

You increase the charging operation type version number if you want to use both the existing set of charging operation types and the updated set of charging operation types. For example, when incrementally upgrading the network mediation software program (client), you may want both the set of charging operation types active at the same time. Both old and new versions of the charging operation types are stored in PDC and published to ECE.

You must update the version number in your mediation specification file to associate the usage request builder with the event definition to which it applies.

The following is the syntax for the charging operation type version:

```xml
<opVersion>
  <externalVersion>version_number</externalVersion>
  <opTypes>
    ...
  </opTypes>
</opVersion>
```

where `version_number` is the version number for the set of charging operation types.

### About Mapping Network Attributes to Event Attributes

The data that ECE needs for charging is sent to ECE as network attributes from clients. For example, the call destination might be sent to ECE as the `Service-Information.PS-Information.Called-Station-Id` attribute. To charge for usage, ECE needs to map network attributes to event attributes. For example, the following example shows how to map the `Service-Information.PS-Information.Called-Station-Id` network attribute to the `CALLED_ID` event attribute.

```xml
<simpleAttributeItem>
  <name>CALLED_ID</name>
  <description>CALLED_ID</description>
  <type>STRING</type>
  <persistedName>my_called_id_c</persistedName>
  <attributeItemSupportedBy>
    <targetApplicationSpecName>Pricing</targetApplicationSpecName>
    <targetApplicationSpecName>Convergent Charging</targetApplicationSpecName>
  </attributeItemSupportedBy>
</simpleAttributeItem>
```

See the discussion about Diameter Gateway in ECE documentation for more information.

### Before Enriching Event Definitions

Before enriching an events definition:
Enriching Event Definitions

To enrich an event definition:

1. Export the event definition from PDC into an XML file by running the following command:

   ```
   ImportExportPricing -export FileNamePrefix -metadata EVENT_ATTRIBUTE_SPEC -n "eventname"
   ```

   where `eventname` is the name of the event to be exported into an XML file.

   For example, this command exports the definition for EventCloudLibrary to the EventCloudLibrary_export_metadata.xml file:

   ```
   ImportExportPricing -export EventCloudLibrary -metadata EVENT_ATTRIBUTE_SPEC -n EventCloudLibrary
   ```

2. Open the event definition file in a text editor.

   The following example shows the event definition for the EventCloudLibrary event:

   ```xml
   <eventAttributeSpec>
     <name>EventCloudLibrary</name>
     <description>EventCloudLibrary</description>
     <internalId>c06de0cc-6105-4cd3-92c9-ad8011e5fe7b</internalId>
     <priceListName>Default</priceListName>
     <obsolete>false</obsolete>
     <attributeSpecSupportedBy>
       <targetApplicationSpecName>Billing</targetApplicationSpecName>
       <targetApplicationSpecName>Pricing</targetApplicationSpecName>
       <targetApplicationSpecName>Convergent Charging</targetApplicationSpecName>
       <targetApplicationSpecName>Realtime Charging</targetApplicationSpecName>
     </attributeSpecSupportedBy>
     <status>ENABLED</status>
     <className>/event/cloud/library</className>
     <complexTypeItem>
       <name>USAGE_INFO</name>
       <description>USAGE_INFO</description>
       <type>STRUCT</type>
       <persistedName>cloudlibrary_usage_info_t</persistedName>
       <simpleAttributeItem>
         ...
       </simpleAttributeItem>
     </complexTypeItem>
   </eventAttributeSpec>
   ```

- Create a new event subclass (for example, `/event/cloud/library`) in Development Center. See BRM Developer’s Guide for more information.

- When you created the event subclass, define a substruct; for example, USAGE_INFO. Add your custom fields to the substruct. Ensure that the fields for storing the user identity and called ID (if you are using zoning in your charge offer) are added to the substruct in the event class.

  See the discussion about substructure in BRM Developer’s Guide for more information.

- The event is mapped to the corresponding service (for example, `service/cloud/library`). See “Setting Up the Service-Event Map” for more information.

- The service and event definitions are synchronized with PDC by running the SyncPDC utility. See “Creating Pricing Setup Components” for more information.
<eventType>SESSION_USAGE</eventType>
<baseEventSpec>EventCloud</baseEventSpec>

<name>MY_IDENTITY</name>
<description>MY_IDENTITY</description>
&type>STRING</type>
<persistentName>my_identity_c</persistentName>
<attributeItemSupportedBy>
  <targetApplicationSpecName>Pricing</targetApplicationSpecName>
  <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
</attributeItemSupportedBy>
</simpleAttributeItem>

<name>MY_CALLED_ID</name>
<description>MY_CALLED_ID</description>
&type>STRING</type>
<persistentName>my_called_id_c</persistentName>
<attributeItemSupportedBy>
  <targetApplicationSpecName>Pricing</targetApplicationSpecName>
  <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
</attributeItemSupportedBy>
</simpleAttributeItem>

<name>SERVICE_PROVIDER</name>
<description>SERVICE_PROVIDER_C</description>
&type>STRING</type>
<persistentName>service_provider_c</persistentName>
<attributeItemSupportedBy>
  <targetApplicationSpecName>Pricing</targetApplicationSpecName>
  <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
</attributeItemSupportedBy>
<length>60</length>
</simpleAttributeItem>

<name>SERVICE_STATUS</name>
<description>SERVICE_STATUS_C</description>
&type>STRING</type>
<persistentName>service_status_c</persistentName>
<attributeItemSupportedBy>
  <targetApplicationSpecName>Pricing</targetApplicationSpecName>
  <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
</attributeItemSupportedBy>
<length>60</length>
</simpleAttributeItem>

<name>SERVICE_TYPE</name>
<description>SERVICE_TYPE_C</description>
&type>STRING</type>
<persistentName>service_type_c</persistentName>
<attributeItemSupportedBy>
  <targetApplicationSpecName>Pricing</targetApplicationSpecName>
  <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
</attributeItemSupportedBy>
<length>60</length>
</simpleAttributeItem>

</complexType>
</complexContent>
</complexType>
</element>
3. Add USER_IDENTITY and CALLED_ID (if you are using zoning in your product offerings) attributes as simple attribute items immediately under the <className> element by doing the following:

a. Search for the following element:

   <className>EventClassName</className>

   where EventClassName is the class name of the event.

   For example:

   <className>/event/cloud/library</className>

b. Add the USER_IDENTITY and CALLED_ID attributes as simple attribute items immediately under the <className> element:

   <className>/event/cloud/EventUsage</className>
   <simpleAttributeItem>
     <name>USER_IDENTITY</name>
     <description>USER_IDENTITY</description>
     <type>STRING</type>
     <persistedName>PersistedNameofAttribute</persistedName>
     <attributeItemSupportedBy>
       <targetApplicationSpecName>Pricing</targetApplicationSpecName>
       <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
     </attributeItemSupportedBy>
   </simpleAttributeItem>
   <simpleAttributeItem>
     <name>CALLED_ID</name>
     <description>CALLED_ID</description>
     <type>STRING</type>
     <persistedName>PersistedNameofAttribute</persistedName>
     <attributeItemSupportedBy>
       <targetApplicationSpecName>Pricing</targetApplicationSpecName>
       <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
     </attributeItemSupportedBy>
   </simpleAttributeItem>

   where PersistedNameofAttribute specifies the name of the BRM database table and column associated with the attributes used for storing the user identity and called ID; for example, cloudlibrary_usage_info_t.my_identity_c and cloudlibrary_usage_info_t.my_called_Id_c.

4. Add the event attributes that you use in ratable usage metric (RUM) expressions for Initiate or Update charging operations as top-level attributes under the <className> element.

   The following example shows the event attributes used in the RUM expressions for charging the EventCloudLibrary event:

   <simpleAttributeItem>
     <name>DURATION</name>
     <description>DURATION</description>
     <type>UNIT_VALUE</type>
<unitType>TimeUnit</unitType>
<attributeItemSupportedBy>
  <targetApplicationSpecName>Pricing</targetApplicationSpecName>
  <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
</attributeItemSupportedBy>
<simpleAttributeItem>
  <name>SPECIFIC_UNIT</name>
  <description>SPECIFIC_UNIT</description>
  <type>UNIT_VALUE</type>
  <unitType>Occurrence</unitType>
  <attributeItemSupportedBy>
    <targetApplicationSpecName>Pricing</targetApplicationSpecName>
    <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
  </attributeItemSupportedBy>
</simpleAttributeItem>

5. Add REQUESTED_UNITS and USED_UNITS attributes as complex attribute items in the file:

<complexType>
  <name>REQUESTED_UNITS</name>
  <description>REQUESTED_UNITS</description>
  <type>STRUCT</type>
  <attributeItemSupportedBy>
    <targetApplicationSpecName>Pricing</targetApplicationSpecName>
    <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
  </attributeItemSupportedBy>
</complexType>
<complexType>
  <name>USED_UNITS</name>
  <description>USED_UNITS</description>
  <type>STRUCT</type>
  <attributeItemSupportedBy>
    <targetApplicationSpecName>Pricing</targetApplicationSpecName>
    <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
  </attributeItemSupportedBy>
</complexType>

6. Copy the event attributes you added in step 4 (if applicable) and paste them in both REQUESTED_UNITS and USED_UNITS complex attribute items:

For example:

<complexType>
  <name>REQUESTED_UNITS</name>
  <description>REQUESTED_UNITS</description>
  <type>STRUCT</type>
  <attributeItemSupportedBy>
    <targetApplicationSpecName>Pricing</targetApplicationSpecName>
    <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
  </attributeItemSupportedBy>
  <simpleAttributeItem>
    <name>DURATION</name>
    <description>DURATION</description>
    <type>UNIT_VALUE</type>
  </simpleAttributeItem>
</complexType>
7. (Optional) Add transient attributes if required for usage charging:

For example:

```xml
<simpleAttributeItem>
    <name>CELL_ID</name>
    <description>CELL_ID</description>
    <type>STRING</type>
    <attributeItemSupportedBy>
        <targetApplicationSpecName>Pricing</targetApplicationSpecName>
        <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
    </attributeItemSupportedBy>
</simpleAttributeItem>
```
8. Add the charging operation types that you intend to use for usage charging and associate them with the group of event attributes that are necessary to perform the charging operation (for example: Initiate, Terminate, Update, and Cancel).

The syntax for adding charging operation types is as follows:

```xml
<opTypes>
  <opType>ChargingOperationType</opType>
  <action>Input</action>
  <item>
    <attributeItemName>EventAttribute</attributeItemName>
    <optional>true|false</optional>
  </item>
</opTypes>
```

where:

- **ChargingOperationType** is the charging operation type to be used. See Table 27–1 for the list of charging operation types that are used in ECE.
- **EventAttribute** is the event attribute that is used as input for performing the charging operation.

The following example shows the charge operation type and required attributes to initiate a charging session for the EventCloudLibrary event:

```xml
<opTypes>
  <opType>Initiate</opType>
  <action>INPUT</action>
  <item>
    <attributeItemName>START_T</attributeItemName>
    <optional>false</optional>
  </item>
  <item>
    <attributeItemName>DURATION</attributeItemName>
    <optional>true</optional>
  </item>
  <item>
    <attributeItemName>END_T</attributeItemName>
    <optional>true</optional>
  </item>
  <item>
    <attributeItemName>USAGE_INFO</attributeItemName>
    <optional>false</optional>
  </item>
  <item>
    <attributeItemName>USAGE_INFOSERVICE_TYPE</attributeItemName>
    <optional>false</optional>
  </item>
  <item>
    <attributeItemName>USAGE_INFOSERVICE_STATUS</attributeItemName>
    <optional>true</optional>
  </item>
  <item>
    <attributeItemName>USAGE_INFOSERVICE_PROVIDER</attributeItemName>
    <optional>true</optional>
  </item>
</opTypes>
```
<item>
  <attributeItemName>USED_UNITS</attributeItemName>
  <optional>false</optional>
</item>

<item>
  <attributeItemName>USED_UNITS.DURATION</attributeItemName>
  <optional>true</optional>
</item>

<item>
  <attributeItemName>USED_UNITS.SPECIFIC_UNIT</attributeItemName>
  <optional>true</optional>
</item>

<item>
  <attributeItemName>REQUESTED_UNITS</attributeItemName>
  <optional>true</optional>
</item>

<item>
  <attributeItemName>REQUESTED_UNITS.DURATION</attributeItemName>
  <optional>true</optional>
</item>

<item>
  <attributeItemName>REQUESTED_UNITS.SPECIFIC_UNIT</attributeItemName>
  <optional>true</optional>
</item>

9. (Optional) Update the charging operation type version number. See "About Charging Operation Type Versions" for more information.

10. Define the network mapping for the events used for charging by doing the following:
    a. Search for the event attribute that you want to map to the network attribute.
    b. Add the following entry:

        ```xml
        <networkAttributeItem>NetworkAttribute</networkAttributeItem>
        
        where NetworkAttribute is the attribute of requests received from the network.
        
        The following example shows the network mapping for the EventCloudLibrary event:
        ```

        ```xml
        <simpleAttributeItem>
        <name>MY_CALLED_ID</name>
        <description>MY_CALLED_ID</description>
        <type>STRING</type>
        <persistedName>my_called_id_c</persistedName>
        <attributeItemSupportedBy>
        <targetApplicationSpecName>Pricing</targetApplicationSpecName>
        <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
        </attributeItemSupportedBy>
        </simpleAttributeItem>

        <networkAttributeItem>Service-Information.PS-Information.Called-Station-Id</networkAttributeItem>
        ```

        ```xml
        
        ...<complexAttributeItem>
        <name>REQUESTED_UNITS</name>
        <description>REQUESTED_UNITS</description>
        ```
Enriching Event Definitions

Enabling Charging for Custom Events

27-11
<attributeItemSupportedBy>
   <targetApplicationSpecName>Pricing</targetApplicationSpecName>
   <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
</attributeItemSupportedBy>

<networkAttributeItem>Multiple-Services-Credit-Control.Used-Service-Un
it.CC-Service-Specific-Units</networkAttributeItem>
</simpleAttributeItem>
</complexTypeItem>
<complexTypeItem>
   <name>USAGE_INFO</name>
   <description>USAGE_INFO</description>
   <type>STRUCT</type>
   <persistedName>cloudlibrary_usage_info_t</persistedName>
   <simpleAttributeItem>
      <name>SERVICE_PROVIDER</name>
      <description>SERVICE_PROVIDER_C</description>
      <type>STRING</type>
      <persistedName>service_provider_c</persistedName>
      <attributeItemSupportedBy>
         <targetApplicationSpecName>Pricing</targetApplicationSpecName>
         <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
      </attributeItemSupportedBy>
      <networkAttributeItem>Multiple-Services-Credit-Control.Used-Service-Un
it.CC-Time</networkAttributeItem>
      <length>60</length>
   </simpleAttributeItem>
   <simpleAttributeItem>
      <name>SERVICE_STATUS</name>
      <description>SERVICE_STATUS_C</description>
      <type>STRING</type>
      <persistedName>service_status_c</persistedName>
      <attributeItemSupportedBy>
         <targetApplicationSpecName>Pricing</targetApplicationSpecName>
         <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
      </attributeItemSupportedBy>
      <networkAttributeItem>Multiple-Services-Credit-Control.Used-Service-Un
it.CC-Time</networkAttributeItem>
      <length>60</length>
   </simpleAttributeItem>
   <simpleAttributeItem>
      <name>SERVICE_TYPE</name>
      <description>SERVICE_TYPE_C</description>
      <type>STRING</type>
      <persistedName>service_type_c</persistedName>
      <attributeItemSupportedBy>
         <targetApplicationSpecName>Pricing</targetApplicationSpecName>
         <targetApplicationSpecName>ConvergentCharging</targetApplicationSpecName>
      </attributeItemSupportedBy>
      <networkAttributeItem>Multiple-Services-Credit-Control.Used-Service-Un
it.CC-Time</networkAttributeItem>
      <length>60</length>
   </simpleAttributeItem>
</complexTypeItem>
11. Save and close the file.
12. Import the XML file into PDC by running the following command:
Editing Event Definitions

To edit an event definition, update the event definition in PDC by using the `ImportExportPricing` utility. When you edit an event definition, ensure that you use the same event type and service type, and update the charging operation type version as appropriate. See "About Charging Operation Type Versions" for more information.

When you create your usage request builder, you must associate it with the service, event type, and charging operation type version in the event definition to which it applies. You may need to restart your charging client to receive usage requests that adhere to the enriched event definition.

### ImportExportPricing -import -metadata FileNamePrefix_export_metadata.xml -ow

For example:

```
ImportExportPricing -import -metadata EventCloudLibrary_export_metadata.xml -ow
```

The EventCloudLibrary event definition is loaded into PDC. The ECE Pricing Updater publishes the enriched event definition to ECE.

13. Update the version number in your mediation specification file to associate the usage request builder with the event definition to which it applies. See the discussion about editing the mediation specification file in *BRM Implementing Charging* for more information.

You may need to restart your charging client to receive usage requests that adhere to the enriched event definition.

**Editing Event Definitions**

To edit an event definition, update the event definition in PDC by using the `ImportExportPricing` utility. When you edit an event definition, ensure that you use the same event type and service type, and update the charging operation type version as appropriate. See "About Charging Operation Type Versions" for more information.

When you create your usage request builder, you must associate it with the service, event type, and charging operation type version in the event definition to which it applies. You may need to restart your charging client to receive usage requests that adhere to the enriched event definition.
This document describes how to use the Pricing Design Center (PDC) Web Service.

Topics in this document:

- **About the PDC Web Service API**
- **About Using the PDC Web Service**
- **About Creating the XML Files**
- **About Using the PDC Web Service to Create or Modify Pricing Components**

See also:

- **Configuring Charge Offers**
- **About Creating Product Offerings**

### About the PDC Web Service API

The PDC Web service application programming interface (API) allows other Oracle applications as well as third-party applications to interact with PDC. The PDC Web service API enables you to create or modify the pricing components in PDC using the pricing components that are created or modified in an external application.

To use the PDC Web service API to create or modify the pricing components in PDC, the external application must call the PDC Web service API through a custom client application. You can use any language that supports Web services to access the API; for example, Java.

### About Using the PDC Web Service

To use the PDC Web service, you must:

- Ensure that the SSL port for the WebLogic Server domain is enabled. See the discussion about enabling the SSL port for the WebLogic Server domain in *PDC Installation Guide* for more information.
- Ensure that the following system properties are set in WebLogic Server:
  - `weblogic.security.TrustKeyStore`
  - `weblogic.security.CustomTrustKeyStoreFileName`
  - `weblogic.security.CustomTrustKeyStorePassPhrase`
  - `weblogic.security.CustomTrustKeyStoreType`
See the discussion about setting the system properties in the WebLogic Server Administration Console Help for more information.

- Ensure that the following BindingProvider properties are set in WebLogic Server:
  - BindingProvider.USERNAME_PROPERTY
  - BindingProvider.PASSWORD_PROPERTY

See the discussion about setting the BindingProvider properties in the WebLogic Server Administration Console Help for more information.

Creating or modifying pricing components by using the PDC Web service involves:

- Creating the XML file containing the pricing components. See "About Creating the XML Files" for more information.
- Using the PDC Web service to create new pricing components or modify the promoted pricing components as defined in the XML file in PDC. See "About Using the PDC Web Service to Create or Modify Pricing Components" for more information.

### About Creating the XML Files

You need to create an XML file containing the pricing components you want to use. You can create:

- One consolidated XML file containing all the pricing components
- Separate XML files for each type of pricing component (for example, charge offers, discount offers, packages, and so on)

The XML files that you create must conform to the format detailed in the XSD files for pricing components.

The PDC Web service uses the PricingGateway.xsd file for creating or modifying pricing components in PDC. The XSD files describe the structure of the XML document. The XML file you create must comply with the structure defined in the XSD. The PricingGateway.xsd file is available at:

http://hostName:sslPortNumber/pdc/PricingGatewayPort?xsd=1

where:

- hostName is the host name of the machine on which PDC is deployed.
- sslPortNumber is the SSL port number of the domain on which PDC is deployed.

### About Using the PDC Web Service to Create or Modify Pricing Components

Use the PDC Web service to create new pricing components or modify the promoted pricing components as defined in the XML file. The PDC APIs are exposed as Web service operations through the PricingGateway.wsdl file. The PricingGateway.wsdl file defines the Web service that can be called as well as the attributes required to call a specific operation.

You can use the PDC Web service to do the following:

- Create the pricing components defined in an XML file in PDC. See "CreatePricing" for more information.
- Modify the promoted pricing components in PDC as defined in an XML file. See "ModifyPricing" for more information.
- Create the pricing components defined in an XML file in PDC and publish the pricing components to the BRM database. See "CreatePricingAndSubmit" for more information.
- Modify the promoted pricing components in PDC as defined in an XML file and publish the pricing components to the BRM database. See "ModifyPricingAndSubmit" for more information.

Use the following sample code to call the PDC Web service:

```java
PricingGatewayPortType pricingGatewayPortType = (PricingGatewayPortType) Service.getPort(new QName("urn:PricingGateWay","PricingGatewayPort"), PricingGatewayPortType.class);

Map<String, Object> rc = ((BindingProvider)pricingGatewayPortType).getRequestContext();
rc.put(BindingProvider.ENDPOINT_ADDRESS_PROPERTY, http://hostName:sslPortNumber/pdc/PricingGatewayPort?WSDL);
rc.put(BindingProvider.USERNAME_PROPERTY, pdcUserName);
rc.put(BindingProvider.PASSWORD_PROPERTY, pdcUserPassword);

UserContextType userContext = new UserContextType();
userContext.setUserid("pdcUserName");
PricingInputXMLType pricingInputXMLType = new PricingInputXMLType();
pricingInputXMLType.setUserContext(userContext);
ByteArrayOutputStream bs = new ByteArrayOutputStream();
ZipOutputStream out = new ZipOutputStream(bs);
FileInputStream fl = null;
try {
    fl = new FileInputStream(new File("pricing.xml"));
    // Set the compression ratio
    out.setLevel(Deflater.BEST_COMPRESSION);
    ZipEntry ze = new ZipEntry("PDC");
    out.putNextEntry(ze);
    byte[] data = new byte[BUFFER_SIZE];
    int count = 0;
    BufferedInputStream in = new BufferedInputStream(fl);
    while ((count = in.read(data, 0, BUFFER_SIZE)) != -1){
        out.write(data, 0, count);
    }
} catch (Exception e) {
    throw e;
} finally {
    if (fl != null) {
        fl.close();
    }
    if (bs != null) {
        bs.flush();
    }
    if (out != null) {
        out.flush();
        out.close();
    }
}
byte[] bytesToRet = null;
```
if (bs != null) {
    bytesToRet = bs.toByteArray();
}
pricingInputXMLType.setXmlBinaryString(bytesToRet);
PDCResponseType pDCResponseType =
pricingGatewayPortType.createPricingAndSubmit(pricingInputXMLType);

CreatePricing

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of pricing component. If the validation is successful, it retrieves the data from the XML file and creates the pricing components in PDC.

The CreatePricing operation does not publish the pricing components to the BRM database.

Syntax

public oracle.communications.brm.pdc.server.service.types.PDCResponseType
createPricing(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws
oracle.communications.brm.pdc.server.service.PricingExceptionResponse;

ModifyPricing

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of pricing component. If the validation is successful, it retrieves the data from the XML file and updates the existing pricing components in PDC.

The ModifyPricing operation does not publish the pricing components to the BRM database.

Syntax

public oracle.communications.brm.pdc.server.service.types.PDCResponseType
modifyPricing(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws
oracle.communications.brm.pdc.server.service.PricingExceptionResponse;

CreatePricingAndSubmit

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of pricing component. If the validation is successful, it retrieves the data from the XML file, creates the pricing components in PDC, and publishes the pricing components to the BRM database.

Syntax

public oracle.communications.brm.pdc.server.service.types.PDCResponseType
createPricingAndSubmit(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws
oracle.communications.brm.pdc.server.service.PricingExceptionResponse;
ModifyPricingAndSubmit

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of pricing component. If the validation is successful, it retrieves the data from the XML file, updates the existing pricing components in PDC, and publishes the pricing components to the BRM database.

**Syntax**

```java
public oracle.communications.brm.pdc.server.service.types.PDCResponseType modifyPricingAndSubmit(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws oracle.communications.brm.pdc.server.service.PricingExceptionResponse;
```
This document describes the profiles used by Pricing Design Center (PDC). It also explains how to create custom profiles for Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:
- About Pricing Profiles
- About Default Pricing Profiles
- About Customizing Pricing Profiles
- Setting Up Custom Pricing Profiles

See also:
- About Creating Product Offerings

About Pricing Profiles

A *pricing profile* specifies the pricing features that are displayed in the PDC UI. You use pricing profiles to adapt the UI to your business needs and to simplify the user experience. For example, if you do not use time to determine how much to charge for the use of your services, you can set up a pricing profile that does not include the time feature.

Pricing profiles are used with pricing, charges, discounts, chargeshares, selectors, and time models. You select a pricing profile for all those components except pricing when creating them. (PDC selects a profile for pricing that matches the pricing profile of the component that contains the pricing, such as a charge.)

A set of default pricing profiles is included with PDC (see "About Default Pricing Profiles"). You can also create custom pricing profiles (see "About Customizing Pricing Profiles").

About Default Pricing Profiles

During PDC installation, depending on whether you selected to support Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) or not, a set of default pricing profiles is loaded in the XML format. The XML files containing the default pricing profiles are located in the `PDC_home/apps/Samples/Examples` directory, where `PDC_home` is the directory in which the PDC software is installed.

Table 29–1 lists the default pricing profiles that are loaded during PDC installation if ECE is selected. These pricing profiles are supported by ECE rating engine:
See "About Customizing Pricing Profiles" for information about modifying the default pricing profiles.

**About Customizing Pricing Profiles**

Typically, you create a custom pricing profile by copying one of the default pricing profiles and then removing features that you do not need from the copy.

---

**Note:** In profile XML files, pricing features are defined in `<capability>` and `<subCapability>` elements.

**Note:** You cannot modify the Subscription pricing profile.

---

For example, if your business does not charge for services by time, you could remove the **TimeModeling** capability from the Online Usage default pricing profile.
The following rules apply for subcapabilities:

- If only one subcapability exists for a particular capability in a target engine profile, the pricing profile must include that subcapability. Do not remove it.

- If multiple subcapabilities exist for a particular capability in a target engine profile, at least one of those subcapabilities must be included in the pricing profile. The custom pricing profile does not, however, have to use the same subcapability as the default pricing profile.

For example, suppose a target engine profile includes Capability X, which has Subcapabilities A, B, C, and D, and a default pricing profile uses Capability X with Subcapabilities A and B. If you customize a copy of that default pricing profile, the copy could include Capability X with Subcapability D. In this case, the custom pricing profile does not use the same subcapabilities as the default pricing profile.

- If you add capabilities to a custom pricing profile, ensure the profile contains capabilities for only one pricing type. See "About Pricing Types" for more information.

- Only capabilities that are supported by all the target engine profiles associated with a pricing profile can be added to a custom pricing profile. See "About the PDC Target Engine Profiles" for a list of capabilities supported by each PDC target engine profile. See "About Associating Custom Pricing Profiles with Target Engine Profiles" for information about associating pricing profiles with target engine profiles.

**WARNING:** After pricing components are associated with a pricing profile, you should not modify that pricing profile. Doing so might invalidate the pricing components that reference the profile.

See "Setting Up Custom Pricing Profiles" for instructions on how to create custom profiles.

**About Pricing Types**

Each pricing profile can support only one pricing type. A **pricing type** performs the following functions:

- Indicates the type of pricing, such as a usage charge or a discount, that can be configured when the pricing profile is associated with a pricing component

- For custom profiles, determines the set of capabilities that are available for inclusion in the profile
In each pricing profile’s XML file, the pricing type is specified in the `<pricingTypeValue>` element.

Table 29–2 shows the pricing types supported by the default pricing profiles:

<table>
<thead>
<tr>
<th>Default Pricing Profile</th>
<th>Supported Pricing Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Usage</td>
<td>USAGE_CHARGE</td>
</tr>
<tr>
<td>Offline Usage</td>
<td>USAGE_CHARGE</td>
</tr>
<tr>
<td>Common Usage</td>
<td>USAGE_CHARGE</td>
</tr>
<tr>
<td>(For BRM) Standard Discount</td>
<td>DISCOUNT</td>
</tr>
<tr>
<td>Billing-time Discount</td>
<td>RECURRING_DISCOUNT</td>
</tr>
<tr>
<td>ChargeShare</td>
<td>CHARGE_SHARE</td>
</tr>
<tr>
<td>Convergent Usage</td>
<td>USAGE_CHARGE</td>
</tr>
<tr>
<td>(For ECE) Standard Discount</td>
<td>DISCOUNT</td>
</tr>
<tr>
<td>Standard Chargesharing</td>
<td>CHARGE_SHARE</td>
</tr>
</tbody>
</table>

**About the PDC Target Engine Profiles**

PDC uses the following target engine profiles:

- **BRM online rating**: Contains all the pricing capabilities (features) supported by the BRM online rating engine. See "BRM Online Rating Target Engine Profile" for a list of the capabilities in this profile.

- **BRM offline rating**: Contains all the pricing capabilities supported by the BRM offline rating engine. See "BRM Offline Rating Target Engine Profile" for a list of the capabilities in this profile.

- **Subscription**: Contains capabilities required to create components supported by the BRM subscription engine, such as recurring charges, one-time charges, rollovers, folds, and bundles. See "Subscription Engine Profile" for a list of the capabilities in this profile.

- **ECE rating**: Contains all the pricing capabilities supported by ECE. See "ECE Rating Target Engine Profile" for a list of the capabilities in this profile.

- **ECE subscription**: Contains capabilities to support product offerings. See "ECE Subscription Engine Profile" for a list of the capabilities in this profile.

**Note:** During PDC installation, all the target engine profiles are loaded irrespective of whether ECE is supported or not.

- If a target engine capability is identified as mandatory in the following tables, it must be included in any pricing profile that is associated with the target engine profile.

  If two or more mandatory capabilities are mutually exclusive, you can remove all but one of them from a custom profile. If you include more than one, you must also include the exclusive constraint (`<exclusiveConstraint>`) between them. See the default pricing profiles for examples.
About Customizing Pricing Profiles

Note: You cannot customize a target engine profile.

BRM Online Rating Target Engine Profile
The name of the BRM online rating target engine profile is RRE_RATING.
The pricing type of all the capabilities in this profile is USAGE_CHARGE (see “About Pricing Types”).
Table 29–3 lists the capabilities contained in this target engine profile.

Table 29–3 Capabilities In the BRM Online Rating Target Engine Profile

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RateplanSelector</td>
<td>Enables you to use a charge selector when configuring a charge. Cannot be used with the EnhancedZoneModeling capability.</td>
</tr>
<tr>
<td>ChargeRateplan</td>
<td>(Mandatory) Enables you to configure charges for a charge offer.</td>
</tr>
<tr>
<td>SubscriberCurrency</td>
<td>(Mandatory) Enables you to specify a currency for a charge. Multicurrency: This subcapability enables you to select one or more currencies in the Currency field in the Create Charge dialog box.</td>
</tr>
<tr>
<td>MultiRumGraph</td>
<td>Enables you to select one or more RUMs in the Measured By field in the Create Charge dialog box.</td>
</tr>
<tr>
<td>AbsoluteRelativeDateTimeRange</td>
<td>Enables users to specify fixed and relative date ranges when configuring a charge.</td>
</tr>
<tr>
<td>EnhancedZoneModeling</td>
<td>Enables you to use zone models and Usage Scenario (USC) selectors when configuring a charge. USCSelector: This subcapability enables you to use USC selectors when configuring a charge. Cannot be used with the RateplanSelector capability.</td>
</tr>
<tr>
<td>TimeModeling</td>
<td>Enables you to use time models when configuring a usage charge. The following subcapabilities specify which attributes of a time model can be used:</td>
</tr>
<tr>
<td></td>
<td>■ MonthOfYear</td>
</tr>
<tr>
<td></td>
<td>■ DayOfWeek</td>
</tr>
<tr>
<td></td>
<td>■ TimeOfDay</td>
</tr>
<tr>
<td></td>
<td>■ DayOfMonth</td>
</tr>
<tr>
<td>UsageBI</td>
<td>(Mandatory if PriceSelector is not included) Enables you to configure one or more balance impacts for a usage charge. CreditLimitCheck: This subcapability enables you to specify whether to continue rating the remaining quantity after the credit limit is exceeded or to search for a charge that impacts a different balance element whose credit limit has not been exceeded and use that charge instead. StepsBasedOnUsageQuantity: This subcapability enables you to create price steps based on event quantity. StepsBasedOnBalanceElement: This subcapability enables you to create price steps based on balance element quantity. SingleRum: This subcapability enables you to select only one ratable usage metric (RUM) in the Measured By field in the Create Charge dialog box. Cannot be used with the PriceSelector capability.</td>
</tr>
</tbody>
</table>
BRM Offline Rating Target Engine Profile

The name of the BRM offline rating target engine profile is BRE_RATING.

This profile contains capabilities for the following pricing types:

- USAGE_CHARGE (see Table 29–4, "Charging Capabilities in the BRM Offline Rating Target Engine Profile")
- DISCOUNT and RECURRING_DISCOUNT (see Table 29–5, "Discounting Capabilities in the BRM Offline Rating Target Engine Profile")
- CHARGE_SHARE (see Table 29–6, "Charge Sharing Capabilities in the BRM Offline Rating Target Engine Profile")

See "About Pricing Types" for more information.

Table 29–4 lists the charging capabilities contained in this target engine profile:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeRateplan</td>
<td>(Mandatory) Enables you to configure charges for a charge offer. <strong>MultiRum:</strong> This subcapability enables you to select one or more RUMs in the <em>Measured By</em> field in the Create Charge dialog box.</td>
</tr>
<tr>
<td>SubscriberCurrency</td>
<td>(Mandatory) Enables you to specify the currency to use in the charge. <strong>SingleCurrency</strong> This subcapability enables you to select only one currency in the <em>Currency</em> field in the Create Charge dialog box.</td>
</tr>
<tr>
<td>AbsoluteDateRange</td>
<td>Enables you to specify only fixed (not relative) date ranges when configuring a charge.</td>
</tr>
<tr>
<td>ZoneModeling</td>
<td>Enables you to use zone models when configuring a charge. Cannot be used with the EnhancedZoneModeling capability.</td>
</tr>
<tr>
<td>EnhancedZoneModeling</td>
<td>Enables you to use zone models, USC selectors, and Access Point Name (APN) selectors when configuring a charge. <strong>USCSelector:</strong> This subcapability enables you to use a USC selector when configuring a charge. <strong>APNSelector:</strong> This subcapability enables you to use an APN selector when configuring a charge. Cannot be used with the ZoneModeling capability.</td>
</tr>
<tr>
<td>TimeModeling</td>
<td>Enables you to use time models when configuring a usage charge. The following subcapabilities specify which attributes of a time model can be used: <strong>Holiday:</strong> Special day calendars <strong>DayOfWeek:</strong> Days of the week <strong>TimeOfDay:</strong> Time of day <strong>UsesEffectivity:</strong> Effective periods</td>
</tr>
</tbody>
</table>
Table 29–5 lists the discounting capabilities contained in this target engine profile:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscountRateplanSelector</td>
<td>Enables you to use a discount selector when configuring a discount.</td>
</tr>
<tr>
<td>DiscountRateplan</td>
<td>(Mandatory) Enables you to configure discounts for a discount offer.</td>
</tr>
<tr>
<td>AbsoluteDateRange</td>
<td>(Mandatory) Enables you to specify fixed date ranges when configuring a discount.</td>
</tr>
<tr>
<td>DiscountOrChargeShareTrigger</td>
<td>Enables you to specify conditions that must be met before a discount can be applied to a charge.</td>
</tr>
<tr>
<td>DiscountOrChargeShareFilter</td>
<td>Enables you to specify criteria that a charge must satisfy to be eligible for a discount. This capability is included in the Standard Discount pricing profile but not in the Billing-time Discount pricing profile.</td>
</tr>
<tr>
<td>EventBasedChargeSelectorSpec</td>
<td>This subcapability supports the use of event attributes to determine whether a charge qualifies for a particular discount.</td>
</tr>
</tbody>
</table>

Table 29–6 lists the charge sharing capabilities contained in this target engine profile:
About Customizing Pricing Profiles

The name of the subscription engine profile is RRE_SUBSCRIPTION. This profile contains capabilities for the PRODUCT_OFFERING pricing type (see Table 29–7).

See "About Pricing Types" for more information.

Table 29–7 lists the product offering capabilities contained in this engine profile:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeOffering</td>
<td>Enables you to configure a charge offer.</td>
</tr>
<tr>
<td>AlterationOffering</td>
<td>Enables you to configure a discount offer.</td>
</tr>
<tr>
<td>DistributionOffering</td>
<td>Enables you to configure a chargeshare offer.</td>
</tr>
</tbody>
</table>

The name of the subscription engine profile is RRE_SUBSCRIPTION. This profile contains capabilities for the PRODUCT_OFFERING pricing type (see Table 29–7).

See "About Pricing Types" for more information.

Table 29–7 lists the product offering capabilities contained in this engine profile:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeOffering</td>
<td>Enables you to configure a charge offer.</td>
</tr>
<tr>
<td>AlterationOffering</td>
<td>Enables you to configure a discount offer.</td>
</tr>
<tr>
<td>DistributionOffering</td>
<td>Enables you to configure a chargeshare offer.</td>
</tr>
</tbody>
</table>

The name of the profile supported by ECE is ECE_RATING. This profile contains capabilities for the following pricing types:

- USAGE_CHARGE (see Table 29–8, "Charging Capabilities in the ECE Rating Target Engine Profile")
- DISCOUNT (see Table 29–9, "Discounting Capabilities in the ECE Rating Target Engine Profile")
- CHARGE_SHARE (see Table 29–10, "Charge Sharing Capabilities in the ECE Target Engine Profile")

See "About Pricing Types" for more information.

Table 29–8 lists the charging capabilities contained in this target engine profile:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeShareRateplan</td>
<td>(Mandatory) Enables you to configure chargeshares for a chargeshare offer.</td>
</tr>
<tr>
<td>AbsoluteDateRange</td>
<td>(Mandatory) Enables you to specify fixed date ranges when configuring a chargeshare.</td>
</tr>
<tr>
<td>DiscountOrChargeShareTrigger</td>
<td>Enables you to specify conditions that must be met before a chargeshare can be applied to a charge.</td>
</tr>
<tr>
<td>DiscountOrChargeShareFilter</td>
<td>Enables you to specify criteria that a charge must satisfy to be eligible for a chargeshare.</td>
</tr>
<tr>
<td>EventBasedChargeSelectorSpec</td>
<td>This subcapability supports the use of event attributes to determine whether a charge qualifies for a particular chargeshare.</td>
</tr>
<tr>
<td>DiscountOrChargeShareBI</td>
<td>(Mandatory) Enables you to configure one or more balance impacts for a chargeshare.</td>
</tr>
</tbody>
</table>
Table 29–8  Charging Capabilities in the ECE Rating Target Engine Profile

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeRateplan</td>
<td>Enables you to configure charges for a charge offer.</td>
</tr>
<tr>
<td>SubscriberCurrency</td>
<td>(Mandatory) Enables you to specify the currency to use in the charge.</td>
</tr>
<tr>
<td></td>
<td><strong>SingleCurrency</strong>: This subcapability enables you to select only one currency</td>
</tr>
<tr>
<td></td>
<td><strong>MultiCurrency</strong>: This subcapability enables you to select multiple currencies.</td>
</tr>
<tr>
<td>MultiRumGraph</td>
<td>Enables you to select one or more RUMs in the <strong>Measured By</strong> field in the</td>
</tr>
<tr>
<td></td>
<td><em>Create Charge</em> dialog box.</td>
</tr>
<tr>
<td>AbsoluteDateRange</td>
<td>Enables you to specify only fixed (not relative) date ranges when configuring a</td>
</tr>
<tr>
<td></td>
<td>charge.</td>
</tr>
<tr>
<td>ZoneModeling</td>
<td>Enables you to use zone models when configuring a charge.</td>
</tr>
<tr>
<td>GenericSelector</td>
<td>Enables you to use generic selectors when configuring a charge.</td>
</tr>
<tr>
<td>TimeModeling</td>
<td>Enables you to use time models when configuring a usage charge.</td>
</tr>
<tr>
<td></td>
<td>The following subcapabilities specify which attributes of a time model can</td>
</tr>
<tr>
<td></td>
<td>be used:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Holiday</strong>: Special day calendars</td>
</tr>
<tr>
<td></td>
<td>- <strong>MonthOfYear</strong>: Months of the year</td>
</tr>
<tr>
<td></td>
<td>- <strong>DayOfWeek</strong>: Days of the week</td>
</tr>
<tr>
<td></td>
<td>- <strong>TimeOfDay</strong>: Time of day</td>
</tr>
<tr>
<td></td>
<td>- <strong>DayOfMonth</strong>: Days of month</td>
</tr>
<tr>
<td>UsageBI</td>
<td>(Mandatory) Enables you to configure one or more balance impacts for a usage</td>
</tr>
<tr>
<td></td>
<td>charge.</td>
</tr>
<tr>
<td></td>
<td><strong>SingleRum</strong>: This subcapability enables you to select only one RUM in the</td>
</tr>
<tr>
<td></td>
<td><strong>Measured By</strong> field in the <em>Create Charge</em> dialog box.</td>
</tr>
<tr>
<td></td>
<td><strong>StepsBasedOnBalanceElement</strong>: This subcapability enables you to create</td>
</tr>
<tr>
<td></td>
<td>price steps based on balance element quantity.</td>
</tr>
<tr>
<td></td>
<td><strong>StepsBasedOnUsageQuantity</strong>: This subcapability enables you to create</td>
</tr>
<tr>
<td></td>
<td>price steps based on event quantity.</td>
</tr>
<tr>
<td></td>
<td><strong>StepsBasedOnBalanceElementWithThreshold</strong>: This subcapability enables</td>
</tr>
<tr>
<td></td>
<td>you to create price steps based on balance element with threshold.</td>
</tr>
<tr>
<td></td>
<td><strong>StepsBasedOnUsageQuantityWithThreshold</strong>: This subcapability enables</td>
</tr>
<tr>
<td></td>
<td>you to create price steps based on event quantity with threshold.</td>
</tr>
</tbody>
</table>

Table 29–9 lists the discounting capabilities contained in this target engine profile:

Table 29–9  Discounting Capabilities in the ECE Rating Target Engine Profile

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DiscountRateplan</td>
<td>(Mandatory) Enables you to configure discounts for a discount offer.</td>
</tr>
<tr>
<td>AbsoluteDateRange</td>
<td>(Mandatory) Enables you to specify fixed date ranges when configuring a</td>
</tr>
<tr>
<td></td>
<td>discount.</td>
</tr>
</tbody>
</table>
Table 29–10 lists the charge sharing capabilities contained in this target engine profile:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeShareRateplan</td>
<td>(Mandatory) Enables you to configure chargeshares for a chargeshare offer.</td>
</tr>
<tr>
<td>AbsoluteDateRange</td>
<td>(Mandatory) Enables you to specify fixed date ranges when configuring a chargeshare.</td>
</tr>
<tr>
<td>DiscountOrChargeShareTrigger</td>
<td>Enables you to specify conditions that must be met before a chargeshare can be applied to a charge.</td>
</tr>
<tr>
<td>DiscountOrChargeShareFilter</td>
<td>Enables you to specify criteria that a charge must satisfy to be eligible for a chargeshare.</td>
</tr>
<tr>
<td>EventBasedChargeSelectorSpec</td>
<td>This subcapability supports the use of event attributes to determine whether a charge qualifies for a particular chargeshare.</td>
</tr>
<tr>
<td>DiscountOrChargeShareBI</td>
<td>(Mandatory) Enables you to configure one or more balance impacts for a chargeshare.</td>
</tr>
</tbody>
</table>

**ECE Subscription Engine Profile**

The name of the ECE subscription engine profile is ECE_SUBSCRIPTION.

This profile contains capabilities for the PRODUCT_OFFERING pricing type (see Table 29–11).

See "About Pricing Types" for more information.

Table 29–11 lists the product offering capabilities in this engine profile:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeOffering</td>
<td>Enables you to configure a charge offer.</td>
</tr>
<tr>
<td>DiscountOffering</td>
<td>Enables you to configure a discount offer.</td>
</tr>
<tr>
<td>ChargeShareOffering</td>
<td>Enables you to configure a chargeshare offer.</td>
</tr>
</tbody>
</table>

**Target Engine Profiles for the Default Pricing Profiles**

The pricing capabilities (features) in a pricing profile are based on the capabilities in the target engine profiles with which it is associated. A pricing profile can be
associated with one or more target engine profiles (see "About Associating Custom Pricing Profiles with Target Engine Profiles").

Table 29–12 shows the default target engine profile or profiles for each default pricing profile:

<table>
<thead>
<tr>
<th>Default Pricing Profile</th>
<th>Default Target Engine Profile</th>
<th>Target Engine Profile Pricing Capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online Usage</td>
<td>RRE_RATING</td>
<td>See Table 29–3, &quot;Capabilities In the BRM Online Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>Offline Usage</td>
<td>BRE_RATING</td>
<td>See Table 29–4, &quot;Charging Capabilities in the BRM Offline Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>Common Usage</td>
<td>RRE_RATING and BRE_RATING</td>
<td>See Table 29–3, &quot;Capabilities In the BRM Online Rating Target Engine Profile&quot; and Table 29–4, &quot;Charging Capabilities in the BRM Offline Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>Standard Discount</td>
<td>BRE_RATING</td>
<td>See Table 29–5, &quot;Discounting Capabilities in the BRM Offline Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>Billing-time Discount</td>
<td>BRE_RATING</td>
<td>See Table 29–5, &quot;Discounting Capabilities in the BRM Offline Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>ChargeShare</td>
<td>BRE_RATING</td>
<td>See Table 29–6, &quot;Charge Sharing Capabilities in the BRM Offline Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>Convergent Usage</td>
<td>ECE_RATING</td>
<td>See Table 29–8, &quot;Charging Capabilities in the ECE Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>Standard Discount</td>
<td>ECE_RATING</td>
<td>See Table 29–9, &quot;Discounting Capabilities in the ECE Rating Target Engine Profile&quot;.</td>
</tr>
<tr>
<td>Standard Chargesharing</td>
<td>ECE_RATING</td>
<td>See Table 29–10, &quot;Charge Sharing Capabilities in the ECE Target Engine Profile&quot;.</td>
</tr>
</tbody>
</table>

About Associating Custom Pricing Profiles with Target Engine Profiles

Before you can publish pricing components that use a custom pricing profile to a rating engine, you must associate the pricing profile with at least one target engine profile. The associated target engine profiles determine which rating or subscription engines the components are published to.

The target engines must support all the capabilities in the pricing profile. See the tables in "About the PDC Target Engine Profiles" for the capabilities supported by each PDC target engine profile.

---

Note: When a component that uses a pricing profile is submitted for publication, PDC validates the pricing profile against its associated target engine profiles to ensure that the capabilities in the pricing profile are supported by the target engines.

---

If you associate a pricing profile with multiple target engine profiles, the pricing profile must contain only capabilities that are supported by all of the associated target engine profiles. See "About Pricing Profiles" for more information.

If a pricing profile is not associated with a target engine profile, any pricing components that use that profile are not published to a target engine.
Setting Up Custom Pricing Profiles

See "Target Engine Profiles for the Default Pricing Profiles" for information about the target engine profiles associated with the default pricing profiles.

See "Associating Custom Pricing Profiles with Target Engine Profiles" for instructions on how to associate pricing profiles with target engine profiles.

About Importing Custom Pricing Profiles into PDC

You define a custom pricing profile in an XML file and use the ImportExportPricing utility to import the XML file into PDC. The XML file containing the custom pricing profile must conform to the format specified in the profiles.xsd file in PDC_home/apps/xsd.

See "Prerequisites for Using the ImportExportPricing Utility" for information about prerequisites for using the ImportExportPricing utility.

Setting Up Custom Pricing Profiles

To set up a custom pricing profile in PDC:

1. Create a custom pricing profile (see "Creating Custom Pricing Profiles").

2. Associate the profile with a target engine profile (see "Associating Custom Pricing Profiles with Target Engine Profiles").

3. Import the profile into PDC (see "Importing Custom Pricing Profiles into PDC").

Creating Custom Pricing Profiles

To create a custom pricing profile:

1. Make a copy of the default pricing profile to customize.
   
   See "About Default Pricing Profiles" for descriptions and locations of the default pricing profiles.

2. Open the copy in a text or XML editor.

3. In the copy, modify the pricing profile name to distinguish the copy from the original:

   <pricingProfile>
   <name>CustomPricingProfileName</name>
   ...

   where CustomPricingProfileName is the new name for the copy.

4. Add or delete pricing capabilities as necessary.

   Note the following:

   ■ If only one subcapability exists for a particular capability in the target engine profile, the pricing profile must include that subcapability.

   ■ If multiple subcapabilities exist for a particular capability in a target engine profile, at least one of the subcapabilities must be included in the pricing profile. The custom pricing profile does not, however, have to use the same subcapability as the default pricing profile.

   See "About Customizing Pricing Profiles" for more information.

5. Save your changes.
Associating Custom Pricing Profiles with Target Engine Profiles

To associate a custom pricing profile with a target engine profile:

1. Open the file containing the custom pricing profile in a text or XML editor.
2. Add the following entry to the <profiles> element:

   
   ```xml
   <profileUsedBy>
   <pricingProfileName>CustomPricingProfile</pricingProfileName>
   <targetEngineProfileName>TargetEngineProfile1</targetEngineProfileName>
   <targetEngineProfileName>TargetEngineProfile2</targetEngineProfileName>
   <priceListName>Default</priceListName>
   </profileUsedBy>
   
   ```

   where:
   - CustomPricingProfile is the name of the pricing profile that you are associating with the target engine profile.
   - TargetEngineProfile# is the name of the target engine profile that you are associating with the custom pricing profile.

   **Note:** For each additional target engine that you want to associate with the pricing profile, add another <targetEngineProfileName> entry to the <profileUsedBy> element.

3. Save and close the file.

See "About Associating Custom Pricing Profiles with Target Engine Profiles" for more information.

Importing Custom Pricing Profiles into PDC

To import a custom pricing profile into PDC, you must have the Pricing Design Admin role.

To import a custom pricing profile into PDC:

1. Verify that the custom pricing profile XML file is complete and follows the guidelines specified in the profiles.xsd file in the PDC_home/apps/xsd directory.
2. Go to the PDC_home/apps/Samples/Examples directory, and enter the following command:

   ```bash
   UNIX
   ImportExportPricing -import -profile ProfileFileName
   ```

   where ProfileFileName specifies the full path and file name of the XML file containing the custom profile data.

   **Note:** To run this command, you must have PDC_home/apps/bin in your PATH environment variable.

For example, the following command imports a custom pricing profile from the MyPricingProfile.xml file into the PDC database in a UNIX environment:

   ```bash
   ImportExportPricing -import -profile /MyPricingProfile.xml
   ```
Importing and Exporting Pricing and Setup Components

This document describes how to import and export Oracle Communications Billing and Revenue Management (BRM) pricing and setup components in Pricing Design Center (PDC).

Topics in this document:

- About Using the XML Pricing Interface to Create Product Offerings
- About the ImportExportPricing Utility
- Exporting Pricing and Setup Components from PDC
- Importing Pricing and Setup Components
- Listing Pricing or Setup Components
- Deleting Components Mastered in BRM
- Creating the Pricing and Setup Components by Using XML Files

See also:

- About Creating Product Offerings

About Using the XML Pricing Interface to Create Product Offerings

As an alternative to using PDC, you can create and modify product offerings in an XML file and then use the ImportExportPricing command-line utility to do the following:

- Import all or some product offerings configured in an XML file into the PDC database.
  The utility creates any new pricing components and modifies any changed components in the PDC database.

- Export all or some product offerings from the PDC database into an XML file for editing.
  If you export the data into an XML file that contains pricing components, the utility overwrites the entire XML file.

To create or modify product offerings in an XML file, you use a text editor or an XML editor. The XML product offerings must follow the format specified in the appropriate XML Schema Definition (XSD) file. See "ImportExportPricing" for more information.
You can also use the **ImportExportPricing** utility to move product offerings from a PDC database to another PDC database.

### About the ImportExportPricing Utility

The **ImportExportPricing** utility is a command-line interface for importing pricing and setup components to and for exporting pricing and setup components from the PDC database by using XML files.

**ImportExportPricing** is a role-based utility that authenticates and authorizes the users based on their role. The role of the users determines what tasks the users can perform and the information they can see.

The users of this utility can have the following roles:

- **Pricing Design Admin**: The user having this role can import and export all the pricing and setup components.
- **Pricing Reviewer**: The user having this role can only export all the pricing and setup components.
- **Pricing Analyst**: The user having this role can import only pricing components and cannot import setup components and profile data. The user can export all the components.

**ImportExportPricing** allows you to filter the data that you want to import to or export from PDC. For example, you can export all pricing and setup components from PDC or specify specific pricing and setup components to export.

You can use **ImportExportPricing** to:

- Export pricing and setup components from the PDC database to an XML file. See "Exporting Pricing and Setup Components from PDC" for more information.
- Import pricing and setup components defined in an XML file into the PDC database. See "Importing Pricing and Setup Components" for more information.
- List pricing or setup components available in the PDC database. See "Listing Pricing or Setup Components" for more information.
- Delete setup components from the PDC database if it is not prerequisite data for any other component. See "Deleting Components Mastered in BRM" for more information.
- Create pricing and setup components in an XML file and load the data into the PDC database. See "Creating the Pricing and Setup Components by Using XML Files" for more information.

### About Dependencies between Pricing and Setup Components

Before importing a pricing component, you must import the setup components that the pricing component references. You must ensure that the setup components referenced by the pricing components are either in the XML file or are already imported into PDC.

For example, to import a charge, you must ensure that the balance elements that it references already exist in PDC.

This section assumes that the following prerequisite data is already available in PDC:

- Service
- Event
• Account Attributes
• Service-Event Map
• Profiles

Table 30–1 lists the prerequisites for importing the pricing components into PDC. Some of the prerequisites are optional depending on your business functions.

<table>
<thead>
<tr>
<th>Pricing Component</th>
<th>Prerequisite Pricing Component</th>
<th>Prerequisite Setup Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discount Filter</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Time Model</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>APN Selector</td>
<td>None</td>
<td>Impact Categories and Zone Models</td>
</tr>
<tr>
<td>Pricing</td>
<td>None</td>
<td>RUMs, Balance Elements, G/L IDs, and Tax Codes</td>
</tr>
<tr>
<td>USC Selector</td>
<td>None</td>
<td>Impact Categories and Zone Models</td>
</tr>
<tr>
<td>Discount Trigger</td>
<td>None</td>
<td>Balance Elements</td>
</tr>
<tr>
<td>Generic Selector</td>
<td>None</td>
<td>Custom Rules and Profile Attribute Specification</td>
</tr>
<tr>
<td>Price Selector</td>
<td>Pricing</td>
<td>None</td>
</tr>
<tr>
<td>Charge</td>
<td>Time Models, Pricing, USC Selectors, APN Selectors, and Price Selectors</td>
<td>RUMs, Balance Elements, Tax Codes, G/L IDs, and Zone Models</td>
</tr>
<tr>
<td>Charge Selector</td>
<td>Charges</td>
<td>Value Maps and Zone Models</td>
</tr>
<tr>
<td>Charge Offer</td>
<td>Charge Selectors and Charges</td>
<td>RUMs, Tax Codes, Tax Suppliers, and Provisioning Tags</td>
</tr>
<tr>
<td>Discount</td>
<td>Pricing, Discount Trigger, and Discount Filter</td>
<td>Balance Elements, Tax Codes, and G/L IDs</td>
</tr>
<tr>
<td>Discount Selector</td>
<td>Discounts</td>
<td>None</td>
</tr>
<tr>
<td>Discount Offer</td>
<td>Discount Selectors and Discounts</td>
<td>Provisioning Tags</td>
</tr>
<tr>
<td>Discount Offer Exclusion</td>
<td>Discount Offer</td>
<td>None</td>
</tr>
<tr>
<td>Bundle</td>
<td>Charge Offers and Discount Offers</td>
<td>None</td>
</tr>
<tr>
<td>Bundle Dependency</td>
<td>Bundles</td>
<td>None</td>
</tr>
<tr>
<td>Bundle Transition</td>
<td>Bundles</td>
<td>None</td>
</tr>
<tr>
<td>Package</td>
<td>Bundles</td>
<td>Balance Elements</td>
</tr>
<tr>
<td>Package Discount Restriction</td>
<td>Bundles</td>
<td>None</td>
</tr>
<tr>
<td>Package List</td>
<td>Packages</td>
<td>None</td>
</tr>
<tr>
<td>Package Transition</td>
<td>Packages</td>
<td>None</td>
</tr>
<tr>
<td>Chargeshare</td>
<td>Pricing, Trigger, Filter, and Charge Selector Spec</td>
<td>Balance Elements, Tax Codes, and G/L IDs</td>
</tr>
</tbody>
</table>
ImportExportPricing uses separate XSD files for importing or exporting pricing components, setup components, and profile data. The XSD files are available in their respective directories in the PDC_home/apps/xsd directory, where PDC_home is the directory in which the PDC software is installed. The XSD files describe the structure of the XML document. The XML file you create must comply with the structure defined in the XSD.

The XSD defines the following items for an XML file:

- The elements and attributes, their data types, and the default and fixed values for the elements and attributes.
- Elements that are child elements, and the number and order of the child elements.
- If an element can be empty or can include text.

Prerequisites for Using the ImportExportPricing Utility

To use ImportExportPricing, you must:

- Ensure that the XML files used by this utility conform to the format detailed in the corresponding XSD files. See "About the XSD Files" for more information.
- Ensure PDC_home/apps/bin is in your PATH environment variable.

Exporting Pricing and Setup Components from PDC

Exporting by using ImportExportPricing involves extracting the pricing and setup components from the PDC database into XML files.

Before exporting pricing or setup components, consider the following:

- You can export all or a subset of pricing and setup components.
- When exporting pricing components, the utility does not export the setup component referenced by pricing components. You must export the setup components explicitly.

ImportExportPricing enables you to export the following from a PDC database to an XML file:

- All pricing and setup components. See "Exporting All the Pricing and Setup Components" for more information.
- A subset of pricing and setup components. See "Exporting a Subset of Pricing and Setup Components" for more information.

Exporting All the Pricing and Setup Components

To export all the pricing and setup components to XML files, enter the following command:

ImportExportPricing
Exporting Pricing and Setup Components from PDC

- `export [FileNamePrefix]`
- `config`
- `pricing`
- `profile`
- `appsvruser ApplicationServerUserName`
- `pdcuser PdcUserName`

See "ImportExportPricing" for a description of parameters used in this command.

For example:

```
ImportExportPricing -export MyData -config -pricing -profile -appsvruser weblogic -pdcuser pdcuser
```

exports all the pricing components, setup components, and profile data in the PDC database in a UNIX environment to the `MyData_export_pricing.xml`, `MyData_export_config.xml`, and `MyData_export_profile.xml` files, respectively.

### Exporting a Subset of Pricing and Setup Components

You can export specific pricing and setup components based on the object type, modification time, and the user who modified the data. You can also filter the exported data based on the object name.

To export a subset of pricing and setup components, enter the following command:

```
ImportExportPricing -export [FileNamePrefix] -config [SetupObjectType1,SetupObjectType2,...] -pricing [PricingObjectType1,PricingObjectType2,...] [-n "ObjectName1, ObjectName2,..."] [-productSpecName ProductSpecName1,ProductSpecName2,...] [-expRefs] [-ma time] [-mby user1, user2,...] -appsvruser ApplicationServerUserName -pdcuser PdcUserName
```

See "ImportExportPricing" for a description of parameters used in this command.

See "About Supported Pricing and Setup Components" about the command line representations for the pricing components and setup components.

### Exporting a Subset of Pricing Components

For example:

```
ImportExportPricing -export MyPricingComponents -pricing ALTERATION_RATE_PLAN, CHARGE_RATE_PLAN -appsvruser weblogic -pdcuser pdcuser
```

exports the ALTERATION_RATE_PLAN and CHARGE_RATE_PLAN pricing components from the PDC database in a UNIX environment to the `MyPricingComponents_export_pricing.xml` file.

### Exporting Only the Setup Components that are Mastered in PDC

For example:

```
ImportExportPricing -export -brmobject
```

exports all the setup components that are mastered in PDC from the PDC database in a UNIX environment to the `MySetupComponents_export_pdc_config.xml` file.
Exporting Only the Setup Components that are Mastered in BRM
For example:  
ImportExportPricing -export -brmobject GLID GLID

exports the G/L IDs into the GLID.XML file.
If you do not specify BRMObjectType, the utility exports all the components mastered in BRM into the XML file.

Exporting All Pricing Reference Objects
For example:  
ImportExportPricing -export MyChargeOffers -pricing  CHARGE_OFFERING -name ChargeOffer-RealtimeUsage, ChargeOffer-Batch -expAllRefs

exports the ChargeOffer-RealtimeUsage, ChargeOffer-Batch including the corresponding metadata and setup components from the PDC database in a UNIX environment to the MyChargeOffers_export_pricing.xml, MyChargeOffers_export_metadata.xml and MyChargeOffers_export_config.xml files.

Exporting Only the Metadata Objects
For example:  
ImportExportPricing -export MySetupComponents -metadata

exports all the event, service, account, and profile attribute specifications from the PDC database in a UNIX environment to the MySetupComponents_export_meta_config.xml file.

Exporting Setup Components Modified After a Specific Time
For example:  
ImportExportPricing -export MySetupComponents -config -ma 2010-01-05T19:05:09GMT+05:30 -appsvruser weblogic -pdcuser pdcuser

exports all the setup components modified after the 2010-01-05T19:05:09GMT+05:30 time from the PDC database in a UNIX environment to the MySetupComponents_export_config.xml file.

Exporting Pricing Components Modified by Specific Users
For example:  
ImportExportPricing -export MyPricingComponents -pricing -mby User1, User2 -appsvruser weblogic -pdcuser pdcuser

exports the pricing components modified by User1 and User2 from the PDC database in a UNIX environment to the MyPricingComponents_export_pricing.xml file.

Exporting Only Successfully Promoted and Transformed Components
You can export only the successfully promoted and transformed versions of the components by running the ImportExportPricing utility. By default, the ImportExportPricing tool exports only successfully promoted and transformed objects. Running the ImportExportPricing utility with the inclFailedObjs parameter exports the most current promoted components in addition to the failed components.
To export only successfully promoted and transformed pricing components, run the following command:

```
ImportExportPricing
    -export
    -pricing
    -name
    -expRefs
    -appsvruser ApplicationServerUserName  
    -pdcuser PdcUserName
```

**Exporting Only Obsoleted Components**

For example:

```
ImportExportPricing -exp -pricing -expObsolete MyPricingObsoleteComponents  
    -appsvruser weblogic -pdcuser pdcuser
```

exports all the obsoleted pricing components (including metadata and setup components) from PDC in a UNIX environment to the `MyPricingObsoleteComponents_export_pricing.xml` file.

**About Supported Pricing and Setup Components**

Table 30–2 lists the type of pricing components and their representation in the command line and the XML file that you can import to or export from PDC by using `ImportExportPricing` utility.

**Table 30–2 Supported Pricing Components and Their Representation in the Command Line and XML File**

<table>
<thead>
<tr>
<th>Pricing Component Type</th>
<th>Represented in the Command Line and XML File As ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge</td>
<td>CHARGE_RATE_PLAN</td>
</tr>
<tr>
<td>Discount</td>
<td>ALTERATION_RATE_PLAN</td>
</tr>
<tr>
<td>Chargeshare</td>
<td>DISTRIBUTION_RATE_PLAN</td>
</tr>
<tr>
<td>Rollover</td>
<td>ROLLOVER_RATE_PLAN</td>
</tr>
<tr>
<td>Discount Selector</td>
<td>ALTERATION_RATE_PLAN_SELECTOR</td>
</tr>
<tr>
<td>Generic Selector</td>
<td>GENERIC_SELECTOR</td>
</tr>
<tr>
<td>Charge Selector</td>
<td>CHARGE_RATE_PLAN_SELECTOR</td>
</tr>
<tr>
<td>Price Selector</td>
<td>PRICE_MODEL_SELECTOR</td>
</tr>
<tr>
<td>APN Selector</td>
<td>APN_MAP</td>
</tr>
<tr>
<td>USC Selector</td>
<td>USC_MAP</td>
</tr>
<tr>
<td>Time Model</td>
<td>TIME_MODEL</td>
</tr>
<tr>
<td>Trigger</td>
<td>TRIGGER_SPEC</td>
</tr>
<tr>
<td>Filter</td>
<td>CHARGE_SELECTOR_SPEC</td>
</tr>
<tr>
<td>Discount Pricing</td>
<td>ALTERATION_POP_MODEL</td>
</tr>
<tr>
<td>Rollover Rules</td>
<td>ROLLOVER_POP_MODEL</td>
</tr>
<tr>
<td>Recurring Pricing</td>
<td>RECURRING_POP_MODEL</td>
</tr>
<tr>
<td>One-Time Pricing</td>
<td>ONE_TIME_POP_MODEL</td>
</tr>
</tbody>
</table>
Table 30–2 (Cont.) Supported Pricing Components and Their Representation in the Command Line and XML File

<table>
<thead>
<tr>
<th>Pricing Component Type</th>
<th>Represented in the Command Line and XML File As ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge Offer</td>
<td>CHARGE_OFFERING</td>
</tr>
<tr>
<td>Discount Offer</td>
<td>ALTERATION_OFFERING</td>
</tr>
<tr>
<td>Chargeshare Offer</td>
<td>DISTRIBUTION_OFFERING</td>
</tr>
<tr>
<td>Discount Exclusion</td>
<td>ALTERATION_EXCLUSION</td>
</tr>
<tr>
<td>Bundle</td>
<td>BUNDLED_PRODUCT_OFFERING</td>
</tr>
<tr>
<td>Bundle Transition</td>
<td>BUNDLE_TRANSITION</td>
</tr>
<tr>
<td>Bundle Dependency</td>
<td>BUNDLE_DEPENDENCY</td>
</tr>
<tr>
<td>Package</td>
<td>PACKAGE_OBJ</td>
</tr>
<tr>
<td>Discount Restriction</td>
<td>PACKAGE_EXCLUSION</td>
</tr>
<tr>
<td>Package Transition</td>
<td>PACKAGE_TRANSITION</td>
</tr>
<tr>
<td>Package List</td>
<td>PACKAGE_LIST</td>
</tr>
</tbody>
</table>

Table 30–3 lists the type of setup components and their representation in the command line and the XML file that you can import to or export from PDC by using ImportExportPricing.

Table 30–3 Supported Setup Components and their Representation in the Command Line and XML File

<table>
<thead>
<tr>
<th>Setup Component Type</th>
<th>Represented in the Command Line and XML File As ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value Map</td>
<td>ZONE_MAP</td>
</tr>
<tr>
<td>Event</td>
<td>EVENT_ATTRIBUTE_SPEC</td>
</tr>
<tr>
<td>Service</td>
<td>PRODUCT_ATTRIBUTE_SPEC</td>
</tr>
<tr>
<td>Account Attribute</td>
<td>CUSTOMER_ATTRIBUTE_SPEC</td>
</tr>
<tr>
<td>RUM</td>
<td>RUM_CONFIGURATION</td>
</tr>
<tr>
<td>Impact Category for Zones</td>
<td>ZONE_RESULT_CONFIGURATION</td>
</tr>
<tr>
<td>Service-Event Map</td>
<td>ATTRIBUTE_SPEC_MAP</td>
</tr>
<tr>
<td>Balance Element</td>
<td>BALANCE_ELEMENT</td>
</tr>
<tr>
<td>G/L ID</td>
<td>GLID</td>
</tr>
<tr>
<td>Tax Code</td>
<td>TAX_CODE</td>
</tr>
<tr>
<td>Tax Supplier</td>
<td>TAX_SUPPLIER</td>
</tr>
<tr>
<td>Geographical Zone Model</td>
<td>GEO_ZONE_MODEL</td>
</tr>
<tr>
<td>Standard Zone Model</td>
<td>STANDARD_ZONE_MODEL</td>
</tr>
<tr>
<td>Provisioning Tag</td>
<td>PROVISIONING_TAG</td>
</tr>
<tr>
<td>Special Day Calendar</td>
<td>HOLIDAY_CALENDAR</td>
</tr>
<tr>
<td>Custom Rule</td>
<td>CUSTOM_ANALYZER_RULE</td>
</tr>
<tr>
<td>Profile Attribute Specification</td>
<td>PROFILE_ATTRIBUTE_SPEC</td>
</tr>
</tbody>
</table>
Importing Pricing and Setup Components

Importing pricing and setup components by using the ImportExportPricing utility involves retrieving data from an XML file and loading it into the PDC database. For the ImportExportPricing utility to import pricing and setup components, the XML file must conform to the format detailed in the XSD files for pricing or setup components. See "About the XSD Files" for more information.

Before importing pricing and setup components, consider the following:

- If you import the pricing and setup components at the same time, the utility imports setup components before importing the pricing components. If the import of setup components is not successful, the utility does not import the pricing components. When importing data, you must specify at least -pricing, -config, or -profile in the command.

- During import, the utility modifies the objects in the database:
  - If the -ow parameter is used and the objects already exist in the database, the utility overwrites the existing objects.
  - If the -ow parameter is not used and the objects already exist in the database, the utility generates an error.
  - If the object does not exist in the database, the utility creates the object, regardless of the usage of the -ow parameter.
  - If the -ignoreID parameter is used, the utility ignores the internal IDs in the specified XML file and imports the components based on the input file name.

To import pricing and setup components, enter the following command:

```
ImportExportPricing
  -config SetupFileName
  -pricing PricingFileName
  -profile ProfileFileName
  -appsvruser ApplicationServerUserName
  -pdcuser PdcUserName
```

See "ImportExportPricing" for a description of parameters used in this command.

For example:

```
ImportExportPricing -import -config /MySetupComponents.xml -appsvruser weblogic -pdcuser pdcuser
```

imports the setup components from the MySetupComponents.xml file to the PDC database in a UNIX environment.

The ImportExportPricing utility imports the data from the XML file into the PDC database and commits the data. If there is a failure, the ImportExportPricing utility rolls back the data and logs the errors in the log file.

Listing Pricing or Setup Components

You can use the ImportExportPricing utility to display the pricing or setup components available in the PDC database.

To list the pricing or setup components, enter the following command:

```
ImportExportPricing
  -t config | pricing
  -appsvruser ApplicationServerUserName
```
Publishing Pricing and Setup Components

You can use the `ImportExportPricing` utility with `-publish` and `-target` parameters to publish setup components from the PDC to the specified target engines, such as batch rating engine, real-time rating engine, and ECE.

To publish setup components, enter the following command:

```
ImportExportPricing
   -publish [Component] [ObjectType1,ObjectType2,...]
   -target Target_Engine
   -appsvruser ApplicationServerUserName
   -pdcuser PdcUserName
```

See "ImportExportPricing" for a description of parameters used in this command.

For example:

```
ImportExportPricing -publish config -target ece
```

All the balance elements in PDC are published to ECE.

Deleting Components Mastered in BRM

```
ImportExportPricing
   -d FilePath
   -appsvruser ApplicationServerUserName
   -pdcuser PdcUserName
```

See "ImportExportPricing" for a description of parameters used in this command.

For example:

```
ImportExportPricing -d /Temp/BRMMasteredComponentsToDelete.xml -appsvruser weblogic -pdcuser pdcuser
```

deletes the components mastered in BRM defined in the
/Temp/BRMMasteredComponentsToDelete.xml file in PDC in a UNIX environment.

Deleting Pricing Profiles

You can also use the `ImportExportPricing` utility to delete pricing profiles if they are not being used by any pricing component.

To delete pricing profiles, enter the following command:

```
ImportExportPricing
   -dp FilePath
   -appsvruser ApplicationServerUserName
   -pdcuser PdcUserName
```

See "ImportExportPricing" for a description of parameters used in this command.

For example:

```
ImportExportPricing -dp /Temp/ProfilesToDelete.xml -appsvruser weblogic -pdcuser pdcuser
```
deletes the pricing profile data defined in the /Temp/ProfilesToDelete.xml file in PDC in a UNIX environment.

Deleting Old Versions of PDC Components

You can delete old versions of PDC components that are obsolete in PDC or in the associated target engines (for example, Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE)) and keep only the latest versions of successfully promoted PDC components.

By running the -keep parameter with the ImportExportPricing utility, you can retain the latest versions of successfully promoted PDC components.

To delete old versions of PDC components, enter the following command:

```bash
ImportExportPricing -keep [metadata | config | pricing | all] -numVersion N -appsvruser ApplicationServerUserName -pdcuser PdcUserName
```

Where N is a positive integer that specifies the number of latest successful versions to be kept in the PDC database.

---

**Note:** PDC keeps N+1 versions of the successfully promoted components if they are available in the PDC database. For example, if you want to keep the latest two versions of successfully promoted components and there is only one version available in the PDC database, PDC keeps only that one version of successfully promoted components.

---

Obsoleting PDC Components

You can use the ImportExportPricing utility to obsolete PDC components in the Promoted status if they are not referenced by any other pricing components.

To obsolete PDC components:

1. Export the PDC component that you want to obsolete into an XML file. See "Exporting Pricing and Setup Components from PDC" for more information.
2. In the XML file, set the <obsolete> element for the PDC component to true:
   ```xml
   <obsolete>true</obsolete>
   ```
3. Import the PDC component into the PDC database with the -ow parameter. See "Importing Pricing and Setup Components" for more information.

   For example:
   ```bash
   ImportExportPricing -import -pricing export_pricing.xml -ow
   ```

   The imported PDC component is obsoleted.

   **Note:** If you want to reuse an obsoleted PDC component, you must set the <obsolete> element for that component to false and import it into the PDC database.
Creating the Pricing and Setup Components by Using XML Files

Ignoring Internal IDs

You can use ImportExportPricing utility with the -ignoreID parameter to ignore the internal IDs and import the pricing and setup components by the input file name.

To ignore internal IDs in the pricing or setup components, enter the following command:

```
ImportExportPricing
  -import config | pricing | SetupFileName|ConfigFileName
  -ignoreID -ow
  -appsvruser ApplicationServerUserName
  -pdcuser PdcUserName
```

See "ImportExportPricing" for a description of parameters used in this command.

For example:

```
ImportExportPricing -import -config pdc_config.xml -ignoreID -ow -pdcuser pdcuser
```

ignores the internal IDs in the `pdc_config.xml` file and imports the setup components into PDC database.

Creating the Pricing and Setup Components by Using XML Files

You can use XML files to create pricing and setup components in the PDC database. Creating the pricing and setup components by using the XML files involves:

- Creating the XML file containing the pricing or setup components. See "Creating the XML Files" for more information.
- Loading the data in the XML file into PDC by using ImportExportPricing. See "Importing Pricing and Setup Components" for more information.

Creating the XML Files

You create the XML file containing the pricing or setup components in an XML editor or a text editor.

You can create:

- One consolidated XML file containing all the pricing or setup components
- Separate XML files for each type of pricing or setup component

The XML files that you create must conform to the format detailed in the XSD files for pricing or setup components. See "About the XSD Files" for more information.

Some pricing components have dependencies on other pricing and setup components. Before loading such pricing components, ensure that the prerequisite data is available in the PDC database. See Table 30–1 for more information.

If you create XML files for specific pricing or setup components, you must load the XML files in the order based on the dependencies. See "About Dependencies between Pricing and Setup Components" for more information.
This document describes how to replicate Pricing Design Center (PDC) data from one Oracle Communications Billing and Revenue Management (BRM) system to another, such as from a development system to a test system.

Topics in this document:
- About Replicating Data
- Replicating Setup Components Mastered in BRM
- Replicating Data Between PDC Databases

See also:
- About Creating Product Offerings

### About Replicating Data

You replicate PDC data by extracting data from a source PDC database and then loading the data into a destination PDC database by using the ImportExportPricing utility. See "Importing and Exporting Pricing and Setup Components" for more information.

The target system must include both the BRM server components and a PDC database. If you create a new BRM installation, do not load the sample files in the PDC_home/apps/samples/examples directory after installing PDC, where PDC_home is the directory in which the PDC software is installed.

Replicating PDC data between two BRM systems involves the following tasks:

1. Replicating Setup Components Mastered in BRM
2. Replicating Data Between PDC Databases

### Replicating Setup Components Mastered in BRM

Some setup components are mastered in BRM and are then synchronized with PDC.

To ensure that all the setup components mastered in the source BRM system are synchronized with the destination BRM and PDC systems, extract the setup components from the source BRM database and load them into the destination BRM database by using the BRM load utilities. When you replicate the PDC data, these components will be synchronized with the destination PDC system.

Table 31–1 lists the type of setup components that are mastered in BRM and their corresponding load utilities.
Table 31–1  Setup Components Mastered in BRM and Their Corresponding Load Utilities in BRM

<table>
<thead>
<tr>
<th>Setup Component Type</th>
<th>BRM Load Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>pin_deploy</td>
</tr>
<tr>
<td>Account</td>
<td>pin_deploy</td>
</tr>
<tr>
<td>Event</td>
<td>pin_deploy</td>
</tr>
<tr>
<td>G/L ID</td>
<td>load_pin_glid</td>
</tr>
<tr>
<td>Tax Supplier</td>
<td>load_tax_supplier</td>
</tr>
<tr>
<td>Provisioning Tag</td>
<td>load_config_provisioning_tags</td>
</tr>
<tr>
<td>Tax Code</td>
<td>NA</td>
</tr>
</tbody>
</table>

Note: To replicate the tax codes, copy the taxcodes_map file in the source BRM system to the destination BRM system. The default location of the taxcodes_map file is the BRM_Home/sys/cm directory, where BRM_Home is the directory in which you installed BRM.

Important: Do not change the internal IDs of the exported attribute specifications.

Replicating Data Between PDC Databases

To replicate data:

1. Extract the service, account, event, and profile attribute specifications from the source PDC database and load them into the destination PDC database by doing the following:

   a. Export the service, account, event, and profile (if applicable) attribute specifications from the source PDC database by running the following command:

      ImportExportPricing -export -metadata

      The service, account, event, and profile (if applicable) attribute specifications are exported to the export_meta_config.xml file.

      Important: Do not change the internal IDs of the exported attribute specifications.

   b. Ensure that the transformation engines are running. See the discussion about starting the transformation engines in PDC Installation Guide for more information.

   c. Import the exported attribute specifications from the export_meta_config.xml file into the destination PDC database by running the following command:

      ImportExportPricing -import -metadata export_meta_config.xml

2. Run the SyncPDC utility by doing the following:

   a. Go to the BRM_Integration_Pack_Home/apps/syncpdc directory.

   b. Enter the following command:

      startSyncPDC
The **Enter Key Password** prompt appears.

c. Enter the password PDC uses for accessing the PDC alias key in the keystore (`BRM_Integration_Pack_Home/apps/conf/pdc.jks`).

3. Extract all the setup components that are defined in PDC from the source PDC database and load them into the destination PDC database by doing the following:

   a. Export all the setup components from the source PDC database by running the following command:

   ```
   ImportExportPricing -export -config
   ```

   The setup components are exported to the `export_config.xml` file.

   b. Import the exported setup components from the `export_config.xml` file into the destination PDC database by running the following command:

   ```
   ImportExportPricing -import -config export_config.xml
   ```

4. Extract all the pricing components from the source PDC database and load them into the destination PDC database by doing the following:

   a. Export all the pricing components from the source PDC database by running the following command:

   ```
   ImportExportPricing -export -pricing
   ```

   The pricing components are exported to the `export_pricing.xml` file.

   b. Import all the pricing components from the `export_pricing.xml` file into the destination PDC database by running the following command:

   ```
   ImportExportPricing -import -pricing export_pricing.xml
   ```
This document provides reference information for Pricing Design Center (PDC) utilities.

Topics in this document:

- ImportExportPricing
- PDCWalletUtil.sh
- PDCBRMWalletUtil.sh
- load_pin_snowball_distribution
- load_pin_spec_rates
- load_pin_sub_bal_contributor
- pin_discount_cleanup
ImportExportPricing

Use the **ImportExportPricing** utility to import and export pricing and setup components from the PDC database by using XML files. See "Importing and Exporting Pricing and Setup Components” for information on using this utility.

The XML files that are used for importing or exporting data must conform to the format detailed in the XML schema definition (XSD) files.

The utility uses separate XSD files for importing or exporting pricing components, setup components, and profile data. The XSD files are available in their respective directories in the `PDC_home/apps/xsd` directory, where `PDC_home` is the directory in which the PDC software is installed. See "About the XSD Files" for more information.

**Location**

`PDC_home/apps/bin`

**Syntax - export**

```bash
importexportpricing
-export [FileNamePrefix]
-config
-config [SetupObjectType1,SetupObjectType2,...]
-metadata
-expObsolete
-pricing [PricingObjectType1,PricingObjectType2,...]
-profile
-ns 'ObjectName1, ObjectName2,...'
-profileSpecName ProductSpecName1,ProductSpecName2,...
[-expRefs -expAllRefs]
[-inclFailedObjs]
[-brmobject [brmObjectType1, brmObjectType2, ...]]
[-pricing [PricingObjectType1,PricingObjectType2,...]]
[-ma time]
[-mby user1, user2,...]
[-l loglevel]
[-v]
[-ow]
[-pricingprofilename 'Name1, Name2,...']
--appsvruser ApplicationServerUserName
-pdcuser PdcUserName
[-h]
```

**Parameters - export**

- **-export [FileNamePrefix]**
  Exports the data from a PDC database and generates a separate XML file for pricing components, setup components, and pricing profile data in the directory from which you run the utility. The utility generates the file names as:
  - `FileNamePrefix_export_pricing.xml` for the file containing pricing components.
  - `FileNamePrefix_export_config.xml` for the file containing setup components.
  - `FileNamePrefix_export_profile.xml` for the file containing pricing profiles data.

If you do not specify `FileNamePrefix`, the utility generates the file name as follows:
- **export_pricing.xml** for the file containing pricing components. If this file already exists in PDC, the utility generates the file name as `export_pricing_timestamp.xml`, where `timestamp` is the server’s local time in the format `yyyy-mm-dd_hh-mm-ss`.

- **export_config.xml** for the file containing setup components. If this file already exists in PDC, the utility generates the file name as `export_config_timestamp.xml`, where `timestamp` is the server’s local time in the format `yyyy-mm-dd_hh-mm-ss`.

- **export_profile.xml** for the file containing pricing profile data. If this file already exists in PDC, the utility generates the file name as `export_profile_timestamp.xml`, where `timestamp` is the server’s local time in the format `yyyy-mm-dd_hh-mm-ss`.

- **-config**
  Exports all the setup components that are defined in PDC from the PDC database to the XML file.

  - **-config [SetupObjectType1, SetupObjectType2,...]**
    Exports `SetupObjectType` from the PDC database to the XML file, where `SetupObjectType` is the setup component type listed in **Table 32–1**.

  **Table 32–1**  Setup Component Types and the Setup Components They Represent

<table>
<thead>
<tr>
<th>Setup Component Type</th>
<th>Setup Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZONE_MAP</td>
<td>Value Map</td>
</tr>
<tr>
<td>EVENT_ATTRIBUTE_SPEC</td>
<td>Event</td>
</tr>
<tr>
<td>PRODUCT_ATTRIBUTE_SPEC</td>
<td>Service</td>
</tr>
<tr>
<td>CUSTOMER_ATTRIBUTE_SPEC</td>
<td>Account Attribute</td>
</tr>
<tr>
<td>RUM_CONFIGURATION</td>
<td>RUM</td>
</tr>
<tr>
<td>ZONE_RESULT_CONFIGURATION</td>
<td>Impact Category for Zones</td>
</tr>
<tr>
<td>ATTRIBUTE_SPEC_MAP</td>
<td>Service-Event Map</td>
</tr>
<tr>
<td>BALANCE_ELEMENT</td>
<td>Balance Element</td>
</tr>
<tr>
<td>GLID</td>
<td>G/L ID</td>
</tr>
<tr>
<td>TAX_CODE</td>
<td>Tax Code</td>
</tr>
<tr>
<td>TAX_SUPPLIER</td>
<td>Tax Supplier</td>
</tr>
<tr>
<td>GEO_ZONE_MODEL</td>
<td>Geographical Zone Model</td>
</tr>
<tr>
<td>STANDARD_ZONE_MODEL</td>
<td>Standard Zone Model</td>
</tr>
<tr>
<td>PROVISIONING_TAG</td>
<td>Provisioning Tag</td>
</tr>
<tr>
<td>HOLIDAYCALENDAR</td>
<td>Special Day Calendar</td>
</tr>
<tr>
<td>CUSTOMIZER_ANALYZER_RULE</td>
<td>Custom Rule</td>
</tr>
<tr>
<td>PROFILE_ATTRIBUTE_SPEC</td>
<td>Profile Attribute Specification</td>
</tr>
</tbody>
</table>

If you do not specify `SetupObjectType`, the utility exports all the setup components from the PDC database to the XML file.

- **-metadata**
  Exports the event, service, account, and profile attribute specifications listed in **Table 32–1** from the PDC database to the XML file.
-expObsolete
Exports all obsolete pricing components (including the corresponding metadata and setup components).

-pricing [PricingObjectType1, PricingObjectType2,...]
Exports PricingObjectType from the PDC database to the XML file, where PricingObjectType is the pricing component type listed in Table 32–2.

Table 32–2 lists the pricing component types and the pricing components they represent.

<table>
<thead>
<tr>
<th>Pricing Component Type</th>
<th>Pricing Component Represented</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARGE_RATE_PLAN</td>
<td>Charge</td>
</tr>
<tr>
<td>ALTERATION_RATE_PLAN</td>
<td>Discount</td>
</tr>
<tr>
<td>DISTRIBUTION_RATE_PLAN</td>
<td>Chargeshare</td>
</tr>
<tr>
<td>ROLLOVER_RATE_PLAN</td>
<td>Rollover</td>
</tr>
<tr>
<td>ALTERATION_RATE_PLAN_SELECTOR</td>
<td>Discount Selector</td>
</tr>
<tr>
<td>GENERIC_SELECTOR</td>
<td>Generic Selector</td>
</tr>
<tr>
<td>CHARGE_RATE_PLAN_SELECTOR</td>
<td>Charge Selector</td>
</tr>
<tr>
<td>PRICE_MODEL_SELECTOR</td>
<td>Price Selector</td>
</tr>
<tr>
<td>APN_MAP</td>
<td>APN Selector</td>
</tr>
<tr>
<td>USC_MAP</td>
<td>USC Selector</td>
</tr>
<tr>
<td>TIME_MODEL</td>
<td>Time Model</td>
</tr>
<tr>
<td>TRIGGER_SPEC</td>
<td>Trigger</td>
</tr>
<tr>
<td>CHARGE_SELECTOR_SPEC</td>
<td>Filter</td>
</tr>
<tr>
<td>ALTERATION_POP_MODEL</td>
<td>Discount Pricing</td>
</tr>
<tr>
<td>ROLLOVER_POP_MODEL</td>
<td>Rollover Rules</td>
</tr>
<tr>
<td>RECURRING_POP_MODEL</td>
<td>Recurring Pricing</td>
</tr>
<tr>
<td>ONE_TIME_POP_MODEL</td>
<td>One-Time Pricing</td>
</tr>
<tr>
<td>CHARGE_OFFERING</td>
<td>Charge Offer</td>
</tr>
<tr>
<td>ALTERATION_OFFERING</td>
<td>Discount Offer</td>
</tr>
<tr>
<td>DISTRIBUTION_OFFERING</td>
<td>Chargeshare Offer</td>
</tr>
<tr>
<td>ALTERATION_EXCLUSION</td>
<td>Discount Exclusion</td>
</tr>
<tr>
<td>BUNDLED_PRODUCT_OFFERING</td>
<td>Bundle</td>
</tr>
<tr>
<td>BUNDLE_TRANSITION</td>
<td>Bundle Transition</td>
</tr>
<tr>
<td>PACKAGE_OBJ</td>
<td>Package</td>
</tr>
<tr>
<td>PACKAGE EXCLUSION</td>
<td>Discount Restriction</td>
</tr>
<tr>
<td>PACKAGE_TRANSITION</td>
<td>Package Transition</td>
</tr>
<tr>
<td>PACKAGE_LIST</td>
<td>Package List</td>
</tr>
</tbody>
</table>

If you do not specify PricingObjectType, the utility exports all the pricing components from the PDC database to the XML file.
-profile
Exports the pricing profile data from the PDC database to the XML file.

-n "ObjectName1, ObjectName2,..."
Filters the pricing or setup components based on the specified component names. You can use an asterisk (*) at the end of the name to search for variations of that name. For example, searching for chargeoffersm* yields variations of that name, such as chargeoffersm1 and chargeoffersm2.

**Note:** When you use the -n parameter, you must specify either -config or -pricing.

The parameters -n and -productSpecName are mutually exclusive.

-productSpecName ProductSpecName1, ProductSpecName2,...
Filters the exported pricing components that are scoped to the specified services.
If you specify this parameter along with PricingObjectType, the utility exports all the pricing component types that are scoped to ProductSpecName. You can use this parameter only for the following pricing component types:

- ALTERATION_OFFERING
- CHARGE_OFFERING
- DISTRIBUTION_OFFERING
- CHARGE_RATE_PLAN

If you specify this parameter without specifying PricingObjectType, the utility exports all pricing components scoped to the specified service.

**Note:** The parameters -n and -productSpecName are mutually exclusive.

-expRefs
Exports all the pricing components referenced by PricingObjectType. For example, if multiple charge offers are referenced by bundles, this parameter exports all the charge offers when exporting bundles. The utility ignores this parameter if it is used with the -config parameter.

-expAllRefs
Exports all the PDC components (including metadata, setup components, and pricing components) referenced by PricingObjectType. For example, if setup components, such as ratable usage metrics (RUMs) and tax codes are referenced by charge offers, then this parameter exports RUMs and tax codes when exporting charge offers. This parameter applies only to the -pricing parameter.

-ma time
Exports pricing components modified on or after time. Enter the time in the format yyyy-mm-dd hh-mm-ss.

-mby user1, user2,...
Exports the pricing components created or modified by the specified users.
-l loglevel
Specifies how much information the utility should log. The logs are stored in the `ImportExportPricing_username_timestamp.log` file, where `username` is the name of user who used the utility and `timestamp` is the time the log file was created in the format `yyyy-mm-dd_hh-mm-ss`.

Set `loglevel` to one of the following:
- FINE
- FINER
- FINEST
- INFO
- OFF
- SEVERE
- WARNING
The default is INFO.

-v
Displays information about successful or failed processing as the utility runs.

-ow
Overwrites any existing XML file containing the exported pricing or setup components with the file specified in the `-export [FileNamePrefix]` parameter.

-pricingprofilename "Name1, Name2,..."
Exports pricing components scoped to the pricing profile names specified in the list.

If this parameter is used with the `-pricing` parameter when a list of pricing component types is specified, the utility exports all the pricing components of the specified type scoped to the pricing profiles names specified in the list.

If this parameter is specified with the `-pricing` parameter when no pricing components types are specified, the utility exports all the pricing components that are scoped to a pricing profile.

-appsvruser ApplicationServerUserName
Specifies the user name to access the WebLogic Server. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

-pdcuser PdcUserName
Specifies the user name to access PDC. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

-h
Displays the syntax for this utility.

**Syntax - import**

```
ImportExportPricing
    -import
    -metadata
```
Parameters - import

- **import**
  Imports the data from the XML file that conforms to the format detailed in the XSD file to the PDC database. You must specify either **-pricing**, **-config**, or **-profile** along with **-import**.

- **metadata**
  Imports the metadata from the metadata filename to the PDC database.

**-config SetupFileName**
Imports the setup components from the **SetupFileName** to the PDC database, where **SetupFileName** specifies the full path and file name of the XML file containing the setup components.

**-pricing PricingFileName**
Imports the pricing components from the **PricingFileName** to the PDC database, where **PricingFileName** specifies the full path and file name of the XML file containing the pricing components.

**-profile ProfileFileName**
Imports the profile data from the **ProfileFileName** file to the PDC database, where **ProfileFileName** specifies the full path and file name of the XML file containing profile data. If you specify **-pricing** and **-config** parameters along with **-profile**, the utility imports profile data and setup components before importing the pricing components. If the import of the profile data and setup components is not successful, the utility does not import pricing components.

- **ignoreID**
  Ignores the internal IDs in the specified XML file and imports the components by the input file name.

- **-l loglevel**
  Specifies how much information the utility should log. The logs are stored in the **ImportExportPricing_username_timestamp.log** file, where **username** is the name of user who used the utility and **timestamp** is the time the log file was created in the format **yyyy-mm-dd hh-mm-ss**.
  Set **loglevel** to one of the following:
  - FINE
  - FINER
  - FINEST
INFO
OFF
SEVERE
WARNING
The default is INFO.

-v
Displays information about successful or failed processing as the utility runs.

-ow
Overwrites any existing pricing or setup components available in the PDC database with the specified components. If the specified pricing or setup component does not exist in PDC, the utility creates new pricing or setup components.

Note: If you do not use the -ow parameter when importing data, the utility generates an error if an object with the same name already exists in PDC.

[-changeSetName ChangeSetName]
Specifies the name of the changeset used to import setup and pricing components into the PDC database. If you do not specify ChangeSetName, the utility generates the changeset name as ieChangeSet:timestamp, where timestamp is the current timestamp in the format yyyy-mm-dd_hh-mm-ss.

-appsvruser ApplicationServerUserName
Specifies the user name to access the WebLogic Server. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

-pdcuser PdcUserName
Specifies the user name to access PDC. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

-h
Displays the syntax for this utility.

Syntax - publish - target
ImportExportPricing
   -publish [Component] [ObjectType1,ObjectType2,...]
   -target Target_Engine

Parameters - publish - target

-publishe [Component]
Publishes the following components from PDC to the specified target engines:

- metadata. Publishes the metadata in PDC to the specified target engine.
- config. Publishes the setup components in PDC to the specified target engine.
- pricing. Publishes the pricing components in PDC to the specified target engine.
- all. Publishes all the components in PDC to the specified target engine.
-**Publish** [**ObjectType1**, **ObjectType2,**…]
Publishes specified components to the target engine; for example, BALANCE_ELEMENT. You can specify [**Component**] or [**ObjectType**] for publishing the components. For the list of component types, see Table 32–1 and Table 32–2.

-**target** [**target engine**]
Specifies the following target engines for publishing the components:
- **Rre.** Publishes the components or object types to the real-time rating target engine.
- **Bre.** Publishes the components or object types to the batch rating target engine.
- **Ece.** Publishes the components or object types to the ECE rating target engine.
- **All.** Publishes the components or object types to all the target engines mentioned above.

-**appsvruser** **ApplicationServerUserName**
Specifies the user name to access the WebLogic Server. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

-**pdcuser** **PdcUserName**
Specifies the user name to access PDC. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

**Syntax - keep**

```
ImportExportPricing
   -keep [metadata | config | pricing | all] -numVersion N
   -appsvruser ApplicationServerUserName
   -pdcuser PdcUserName
```

**Parameters - keep**

-**keep metadata | config | pricing | all**
Deletes old versions of PDC components that are obsolete in PDC or the associated target engines (for example, Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE)) and keeps only the latest versions of successfully promoted PDC components.

-**numVersion N**
Keeps only the latest specified version of all the successfully promoted objects in the PDC database and deletes the remaining versions, where N is a positive integer that specifies the number of latest successful versions to be kept in the PDC database.

-**appsvruser ApplicationServerUserName**
Specifies the user name to access the WebLogic Server. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

-**pdcuser PdcUserName**
Specifies the user name to access PDC. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

**Syntax - delete**

```
ImportExportPricing
   [-d FilePath] [-dp FilePath]
   -appsvruser ApplicationServerUserName
```
Parameters - delete

- **d FilePath**
  Deletes the setup components specified in `FilePath` from the PDC database, where `FilePath` specifies the full path and file name of the XML file containing the setup components. The utility displays an error message if the data being deleted is dependent on other data.

- **dp FilePath**
  Deletes the pricing profiles specified in `FilePath` from the PDC database provided the profiles are not being used by any pricing component, where `FilePath` specifies the full path and file name of the XML file containing the pricing profiles.

- **appsvruser ApplicationServerUserName**
  Specifies the user name to access the WebLogic Server. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

- **pdcuser PdcUserName**
  Specifies the user name to access PDC. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

Syntax - type

```
ImportExportPricing
   -t config | pricing
   -appsvruser ApplicationServerUserName
   -pdcuser PdcUserName
```

Parameters - type

- **t config | pricing**
  Displays the list of the setup or pricing component types supported in PDC.

- **appsvruser ApplicationServerUserName**
  Specifies the user name to access the WebLogic Server. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

- **pdcuser PdcUserName**
  Specifies the user name to access PDC. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

Parameters - ignoreID

- **ignoreID**
  Ignores the internal IDs in the specified XML file and imports the components by the input file name.

- **appsvruser ApplicationServerUserName**
  Specifies the user name to access the WebLogic Server. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.

- **pdcuser PdcUserName**
  Specifies the user name to access PDC. If you do not specify this parameter, the utility prompts you to enter the user name at the command prompt.
Results

The **ImportExportPricing** utility notifies you when it runs successfully. Otherwise, look in the **ImportExportPricing_username_timestamp.log** file, where *username* is the name of user who used the utility and *timestamp* is the timestamp of the log file in the format *yyyy-mm-dd_hh-mm-ss*. This file is in the directory from which the utility was started.
PDCWalletUtil.sh

Use the PDCWalletUtil.sh utility to create the PDC wallet and to change passwords stored in that wallet; for example, WebLogic Server domain, PDC user, and database passwords.

For more information, see "Changing the Password in the PDC Wallet" in BRM System Administrator’s Guide.

Location

PDC_home/apps/bin

Syntax

To create the PDC wallet:

./PDCWalletUtil.sh create walletlocation walletpassword

where:

- **walletlocation** is the location in which the PDC wallet must be created.
- **walletpassword** is the PDC wallet password.

To change the passwords stored in the PDC wallet:

./PDCWalletUtil.sh set walletlocation walletpassword configentry password

where:

- **configentry** is the configuration entry to store the password. The valid values are:
  - `PDC_APP_SERVER_USER_PASSWORD`. The Oracle WebLogic server user password.
  - `PDC_APP_USER_PASSWORD`. The PDC user password.
  - `TARGET_PDC_ADMIN_PASSWORD`. The password of the Oracle WebLogic server user in the target system.
  - `SOURCE_PDC_ADMIN_PASSWORD`. The password of the Oracle WebLogic server user in the source system.
  - `TARGET_PDC_USER_PASSWORD`. The password of the PDC user in the target system.
  - `SOURCE_PDC_USER_PASSWORD`. The password of the PDC user in the source system.
- **password** is the password to be stored.

Parameters

- **create**
  Creates the PDC wallet in the specified location.

Results

The PDCWalletUtil.sh utility notifies you when it successfully creates the wallet or stores the password in the PDC wallet.
PDCBRMWalletUtil.sh

Use the PDCBRMWalletUtil.sh utility to create the BRM Integration Pack wallet and to change passwords stored in that wallet; for example, transformation cross-reference database, migration cross-reference database, and BRM database passwords.

For more information, "Changing the Password in the BRM Integration Pack Wallet" in BRM System Administrator’s Guide.

Location

BRM_Integration_Pack_home/apps/bin

Syntax

To create the BRM Integration Pack wallet:

```
./PDCBRMWalletUtil.sh create walletlocation walletpassword
```

where:

- `walletlocation` is the location in which the BRM Integration Pack wallet must be created.
- `walletpassword` is the BRM Integration Pack wallet password.

To change the passwords stored in the BRM Integration Pack wallet:

```
./PDCBRMWalletUtil.sh set walletlocation walletpassword configentry password
```

where:

- `configentry` is the configuration entry to store the password. The valid values are:
  - `MIGRATION_DB_PASSWORD`. The migration cross-reference database user password.
  - `TRANS_XREF_DB_PASSWORD`. The transformation cross-reference database user password.
  - `BRM_DB_PASSWORD`. The BRM database user password.
- `password` is the password to be stored.

Parameters

`create`

Creates the BRM Integration Pack wallet in the specified location.

Results

The PDCBRMWalletUtil.sh utility notifies you when it successfully creates the wallet or stores the password in the BRM Integration Pack wallet.
load_pin_snowball_distribution

Use the load_pin_snowball_distribution utility to load snowball discount distribution rules into the /config/snowball_distribution object in the BRM database. You define how snowball discounts are distributed in the pin_snowball_distribution file in BRM_home/sys/data/pricing/example.

Caution: The load_pin_snowball_distribution utility overwrites existing distribution rules. If you are updating distribution rules, you cannot load new distribution rules only. You must load complete sets of distribution rules each time you run the load_pin_snowball_distribution utility.

Important: To connect to the BRM database, the load_pin_snowball_distribution utility needs a configuration file in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in BRM System Administrator’s Guide.

Location

BRM_home/bin

Syntax

load_pin_snowball_distribution [-d] [-v] pin_snowball_distribution_file

Parameters

-d
Creates a log file for debugging purposes. Use this parameter for debugging when the utility appears to have run with no errors, but the data has not been loaded into the database.

-v
Displays information about successful or failed processing as the utility runs.

Note: This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter -v at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace filename.log with the name of the log file:

load_pin_snowball_distribution other_parameter -v > filename.log

pin_snowball_distribution_file
The name and location of the file that defines the snowball distribution rules. The default pin_snowball_distribution file is in BRM_home/sys/data/pricing/example.
If you do not run the utility from the directory in which the file is located, you must include the complete path to the file, for example:

```
load_pin_snowball_distribution BRM_home/sys/data/pricing/example
```
load_pin_spec_rates

Use this utility to set up customized rating by loading the contents of the pin_spec_rates file into the BRM database.

For information about customizing rating, see BRM Opcode Guide.

**Caution:** The load_pin_spec_rates overwrites the existing setup for administrative events charges. If you are updating a set of administrative events charges, you cannot load new charges only. You load complete sets of charges each time you run the load_pin_spec_rates utility.

**Note:** To connect to the BRM database, the load_pin_spec_rates utility needs a configuration file in the directory from which you run the utility. See the discussion of configuration files in BRM System Administrator’s Guide.

**Location**

BRM_home/bin

**Syntax**

load_pin_spec_rates pin_spec_rates_file

**Parameters**

*pin_spec_rates_file*

The name and location of the file that maps opcodes to event types to be rated. The default is BRM_home/sys/data/config/pin_spec_rates.

If you copy the pin_spec_rates file to the same directory from which you run the load_pin_spec_rates utility, you do not have to specify either the path or the file name.

**Results**

If the load_pin_spec_rates utility does not notify you that it was successful, look in the default.pinlog file to find any errors. This file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.

**Important:** You must restart the Connection Manager for the new administrative event charges to take effect.
load_pin_sub_bal_contributor

Use this utility to load the configuration for contributor-based sub-balances into the BRM database. You define the sub-balance configuration in the `BRM_home/sys/data/pricing/example/pin_sub_bal_contributor` file.

For information about contributor-based sub-balances, see "About tracking sub-balances".

---

**Caution:** The `load_pin_sub_bal_contributor` utility overwrites the existing sub-balance configurations. If you are updating a set of sub-balance configurations, you cannot load new configurations only. You load complete sets of sub-balance configurations each time you run the `load_pin_sub_bal_contributor` utility.

---

**Important:** To connect to the BRM database, the `load_pin_sub_bal_contributor` utility needs a configuration file in the directory from which you run the utility. See the discussion of configuration files in `BRM System Administrator's Guide`.

---

**Location**

`BRM_home/bin`

**Syntax**

```
load_pin_sub_bal_contributor [-d] [-v] pin_sub_bal_contributor_file
```

**Parameters**

- `-d`  
  Creates a log file for debugging purposes. Use this parameter for debugging when the utility appears to have run with no errors, but the sub-balance configurations have not been loaded into the database.

- `-v`  
  Displays information about successful or failed processing as the utility runs.

---

**Note:** This parameter is always used with other parameters and commands. It is not position dependent. For example, you can enter `-v` at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace `filename.log` with the name of the log file:

```
load_pin_sub_bal_contributor other_parameter -v > filename.log
```

**pin_sub_bal_contributor_file**

The name and location of the file that defines the configuration for contributor-based sub-balances. The default `pin_sub_bal_contributor_file` is in `BRM_home/sys/data/pricing/sample`. 
If you copy the `pin_sub_bal_contributor` file to the directory from which you run the `load_pin_sub_bal_contributor` utility, you do not have to specify either the path or the file name.

**Results**

The `load_pin_sub_bal_contributor` utility notifies you when it successfully creates the `/config/sub_bal_contributor` object.

If the `load_pin_sub_bal_contributor` utility does not notify you that it was successful, look in the `default.pinlog` file to find any errors. This log file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.

`---

**Important:** You must restart the Connection Manager to make the new sub-balance configurations available.

---`
pin_discount_cleanup

Use this utility to change the status of expired discounts from active to canceled.
You use this utility when you set discount midcycle purchase or cancel options that
grant a full discount when the discount is canceled in the middle of an accounting
cycle.
You can run this utility nightly or add it to the pin_bill_day script to be run
automatically. See “Running Billing Utilities” in BRM Configuring and Running Billing.

Location

BRM_home/bin
where BRM_home is the directory in which the BRM server software is installed.

Syntax

pin_discount_cleanup -m close|delete -n days -d date [-v] [-t] [-help]

Parameters

-m close|delete
Specifies whether to delete discounts when they are canceled:
  ■ close
    Changes the status of all active, expired discounts to canceled without deleting the
discounts.
  ■ delete
    Deletes all expired discounts.

-n
The number of days prior to -d date for which to cancel expired discounts. This is used
in conjunction with the -d parameter to determine the actual dates for which discounts
are canceled.

-d
The end date (in the format MM/DD/YYYY) for which discounts are canceled. This is
used in conjunction with the -n parameter to determine the actual dates for which
discounts are canceled.

For example, if -n is 5 (days) and -d is 07/15/2007, discounts that expired from
7/10/2007 are canceled.

Note:
The expire date cannot be greater than the current date. For instance, in the
example above, if 7/10/2007 is greater than the current date, pin_discount_cleanup
returns an error. Similarly, if only -d date is specified, and date is
greater than the current date, pin_discount_cleanup returns an error.
If neither -n or -d parameters are specified, then the current time is used.

-v
Displays information about successful or failed processing as the utility runs.
**Note:** This parameter is always used in conjunction with other parameters and commands. It is not position dependent. For example, you can enter -v at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace filename.log with the name of the log file:

```
pin_discount_cleanup other_parameters -v > filename.log
```

-t  
Displays the number of records processed (the number of discounts that were canceled).

-**help**  
Displays the syntax and parameters for this utility.

**Results**

To check results of running this utility, look in the log file (normally `default.pinlog`) for error messages. The log file is located in the directory from which the utility was started or in a directory specified in the utility’s configuration file (`pin.conf`).
This document lists changes in pricing terminology for this release of Oracle Communications Billing and Revenue Management (BRM) and Pricing Design Center (PDC).

See also:

- About Creating Product Offerings
New Pricing Terminology

Pricing terminology changed in this BRM release. Table A–1 lists those differences.

<table>
<thead>
<tr>
<th>Previous Term</th>
<th>New Term</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChargeShare</td>
<td>chargeshare offer</td>
</tr>
<tr>
<td>ChargeShare model</td>
<td>chargeshare</td>
</tr>
<tr>
<td>deal</td>
<td>bundle</td>
</tr>
<tr>
<td>discount</td>
<td>discount offer</td>
</tr>
<tr>
<td>discount master</td>
<td>discount filter</td>
</tr>
<tr>
<td>discount model</td>
<td>discount</td>
</tr>
<tr>
<td>EVAL(&quot;function_name&quot;) discount expression</td>
<td>Function[&quot;function_name&quot;] discount expression</td>
</tr>
<tr>
<td>plan</td>
<td>package</td>
</tr>
<tr>
<td>plan list</td>
<td>package list</td>
</tr>
<tr>
<td>product</td>
<td>charge offer</td>
</tr>
<tr>
<td>rate plan</td>
<td>charge</td>
</tr>
<tr>
<td>rate plan selector</td>
<td>charge selector</td>
</tr>
<tr>
<td>resource</td>
<td>balance element (design time) or balance (runtime)</td>
</tr>
<tr>
<td>subscription group</td>
<td>service group</td>
</tr>
</tbody>
</table>