# Contents

## Preface

<table>
<thead>
<tr>
<th>Audience</th>
<th>xi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downloading Oracle Communications Documentation</td>
<td>xi</td>
</tr>
<tr>
<td>Documentation Accessibility</td>
<td>xi</td>
</tr>
<tr>
<td>Document Revision History</td>
<td>xi</td>
</tr>
</tbody>
</table>

## 1 List of Customer Management Features

| Customer Management Features    | 1-1 |

## 2 Choosing Account Creation Options

| About creating customer accounts | 2-1 |
| About Customer Information Collected When an Account is Created | 2-1 |
| Specifying How to Validate Customer Contact Information | 2-2 |
| Specifying Defaults for Account Currency, Country, and Email | 2-2 |
| Allowing Accounts to Be Created with Backdated Services or Balances | 2-3 |
| Customizing Account Creation   | 2-3 |

## 3 Collecting Information About Customers

| Making Notes about Customers    | 3-1 |
| Collecting a Summary of Activities Performed on an Account | 3-2 |
| Customizing and Localizing News Feed | 3-2 |
| Customizing the List of Reasons for Account Changes | 3-3 |
| Maintaining an Audit Trail of BRM Activity | 3-4 |

## 4 Setting Up Customer Self Care with Self-Care Manager

| About Self-Care Manager         | 4-1 |
| Supported Application Servers   | 4-1 |
| About Customizing Self-Care Manager | 4-2 |
| About Currencies                | 4-2 |
5 Setting up Welcome Messages to Customers

- Sending a Welcome Email
  - Changing the Welcome Email Subject Line
  - Changing the Welcome Email Sender Address
  - Specifying the Welcome Email Location
  - Disabling the Welcome Email
- Setting Up a Welcome Web Page
- Using Variables to Include Customer Data in Welcome Emails and Web Pages
- Using Multiple Welcome Emails and Web Pages

6 Sending Email to Customers Automatically

- Configuring Email Data Manager
- Starting the Email Data Manager
- Configuring sendmail Options

7 Managing Passwords and Login Names

- About Login Names and Passwords
  - About Telco Service Logins and Passwords
- Using Security Questions
- Assigning Passwords Automatically
- Defining Email Login Names
- Detecting Duplicate Logins

8 Recording Login Failures

- Recording Login Failures
- Specifying the Account That Records Login Failures
- Viewing Login Failures

9 Managing Account and Service Status

- Automatically Inactivating and Closing Accounts
- Scheduling Status Changes in Advance
- Changing Start Times for Reactivated Charge and Discount Offers
- Allowing Active Services with Inactive Accounts
- Inactivating and Closing Accounts in Hierarchies
- Reusing Login Names and Passwords from Closed Accounts or Canceled Services
## Managing Credit Cards During Account Creation

1. Charging Customers at Account Creation 10-1
2. Which Fees Are Charged at Account Creation 10-2
3. How General Ledger Revenue Is Reported for Account Creation Charges 10-2
4. Validating Credit Cards at Account Creation 10-2
5. Avoiding Credit Card Revalidation 10-3
6. Allowing Account Creation without a Credit Card 10-3
7. Increasing Account Creation Speed When Paymentech Is Offline 10-3
8. Specifying the Account That Records Credit Card Validations 10-4

## Changing a Customer's Billing Configuration

1. Changing a Customer's Billing Day of Month 11-1
2. Changing a Customer's Accounting Type 11-1
3. Changing an Account's Bill Unit to Nonpaying 11-2
4. Changing a Closed Child Account's Bill Unit to Nonpaying 11-2

## Managing Purchased Charge Offers

1. Canceling Charge Offers 12-1
2. Canceling Charge Offers Without Charging a Cancel Fee 12-2
3. Including Event Adjustments in Charge Offer Cancellation Refunds 12-2
4. Calculating Conditional Discount Offers When Canceling a Charge Offer 12-3
5. Deleting Canceled Charge Offers 12-3
6. Enabling Charge Offer Purchases from Closed or Inactive Accounts 12-4
7. Changing Charge Offer Quantity 12-5

## Managing Purchased Discount Offers

1. Setting Discount Offer Status 13-1
2. Setting Discount Offer Purchase, Cycle, and Usage Start and End Times 13-1
3. Enabling Mutually Exclusive Discount Offers 13-2
4. Configuring Remaining Charge Cycle Discounting 13-3
5. Canceling Discount Offers 13-4
6. Configuring Discount End Dates during Mid-Cycle Cancellations 13-5
7. Rating Delayed Events for a Canceled Discount Offer 13-6
8. Changing the Status of Discounts Canceled in Mid-Cycle 13-6
9. Deleting Canceled Discount Offers 13-6
<table>
<thead>
<tr>
<th>14</th>
<th>Managing Purchased Bundles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Managing Purchased Bundles</td>
</tr>
<tr>
<td></td>
<td>Modifying Bundles</td>
</tr>
<tr>
<td></td>
<td>Configuring Bundle Dependencies</td>
</tr>
<tr>
<td></td>
<td>Bundle Dependency Validations Involving Inactive or Canceled Charge Offers or Discount Offers</td>
</tr>
<tr>
<td></td>
<td>Enabling Bundle Dependency Validations for Inactive or Canceled Offers</td>
</tr>
<tr>
<td></td>
<td>Canceling Bundles</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>15</th>
<th>Configuring Bundle and Package Transitions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configuring Bundle and Package Transitions</td>
</tr>
<tr>
<td></td>
<td>Transitioning Bundles</td>
</tr>
<tr>
<td></td>
<td>Transitioning Packages</td>
</tr>
<tr>
<td></td>
<td>About Defining Package Transition Rules</td>
</tr>
<tr>
<td></td>
<td>Defining a Generation Change for Packages</td>
</tr>
<tr>
<td></td>
<td>Configuring Services for a Generation Change</td>
</tr>
<tr>
<td></td>
<td>Creating Custom Transition Types for Bundles and Packages</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>16</th>
<th>Backdating Subscription Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Backdating Subscription Actions</td>
</tr>
<tr>
<td></td>
<td>About Backdating Subscription Actions</td>
</tr>
<tr>
<td></td>
<td>About Backdated Status Change</td>
</tr>
<tr>
<td></td>
<td>How Cycle Fees are Calculated for Backdated Status Change</td>
</tr>
<tr>
<td></td>
<td>About Backdating Beyond the G/L Posting Date</td>
</tr>
<tr>
<td></td>
<td>How Effective Time Is Used to Validate Backdating Operations</td>
</tr>
<tr>
<td></td>
<td>Rerating of Events for the Backdated Period</td>
</tr>
<tr>
<td></td>
<td>About Backdated Charge Offer, Discount Offer, or Bundle Cancellation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>17</th>
<th>Creating Custom Service Life Cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creating Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>About Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>Triggering State Changes in Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>Configuring Business Rules for Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>About Creating Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>Creating Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>Modifying Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>Enabling BRM to Use Custom Service Life Cycles</td>
</tr>
<tr>
<td></td>
<td>Adding SLM Entries to the CM pin.conf File</td>
</tr>
<tr>
<td></td>
<td>Mapping States to Statuses</td>
</tr>
<tr>
<td></td>
<td>About the Default State of a Status</td>
</tr>
<tr>
<td></td>
<td>About the Service State Mapping Configuration File</td>
</tr>
<tr>
<td></td>
<td>Mapping States to Statuses</td>
</tr>
</tbody>
</table>
Modifying State-to-Status Mapping 17-10
Deleting State-to-Status Mapping 17-11
Associating Services with Custom Life Cycles 17-12
Associating Bill Units with the SLM Business Profile 17-13

18 Creating and Managing Service Groups

About Grouping Services by Subscription 18-1
About Service Groups 18-2
  Sharing Charge and Discount Offers 18-2
  Sharing Balance Groups 18-2
  About Using Charge and Discount Offers to Rate Service Group Usage 18-3
  About Distributing Discounts to Member Services 18-3
    Preventing Member Services from Inheriting Billing-Time Discounts 18-4
  About Sharing Charges and Discounts in Service Groups 18-4
    About Distributing Sharing Group Balances to Member Services 18-4
  About Changing Status in Service Groups 18-6
  About Setting Up Corporate Accounts That Use Service Groups 18-6
About Creating and Managing Service Groups 18-7
  About Creating a Service Group 18-7
  About Adding Bundles to Service Groups 18-7
  About Setting Up Balance Groups for Service Groups 18-8
  About Sharing ERAs in Service Groups 18-10
  About Modifying Service Groups 18-10
  About Canceling a Subscription Service 18-11
    About Service Status When a Subscription Service Is Canceled 18-11
    Effect of Canceling a Subscription Service on Offers Owned by the Service 18-11
    Effect of Canceling a Subscription Service on Sharing Groups Owned by the Service 18-12
  About Transferring a Service Group to Another Account 18-12
    About Transferring Service Groups between Schemas 18-13
    About Transferring Service Groups Associated with Objects 18-13
    About Transferring Balance Groups during Service Group Transfer 18-15
    About Transferring Noncurrency Assets during Service Group Transfer 18-16
    About Extending Sub-Balance Validity When a Service Group Is Transferred 18-16
    About Consuming Noncurrency Balances When a Service Group Is Transferred 18-17
    About Rolling Over Noncurrency Balances When a Service Group Is Transferred 18-20
    About Prorating Cycle Fees When a Service Group Is Transferred 18-21
    Applying Billing-Time Discounts and Folds When a Service Group Is Transferred 18-22
18-22 About Rating Delayed Events When a Service Group Is Transferred
18-23 How a Service Group Transfer Affects Account Migration
18-23 Configuring Sub-Balance Validity for Service Group Transfer
18-24 About Billing for Service Group Usage
18-24 About Creating Multiple Bills for a Service Group
18-25 About Billing for Multiple Service Groups
18-25 About Billing on Subscription Service Cancellation
18-25 About Billing an Account When a Service Group Is Canceled during the Delayed Billing Period
18-25 About Billing a Service Group after Transferring It to Another Account

### 19 Creating and Managing Profile Sharing Groups

19-1 About Profile Sharing Groups
19-2 About Profile Sharing Group Membership
19-3 Creating Profile Sharing Groups
19-3 Adding a Profile Group to a Member's Ordered Balance Group
19-4 Modifying Profile Sharing Groups
19-4 Deleting Profile Sharing Groups

### 20 Managing System and Account Currencies

20-1 About System and Account Currencies
20-2 Setting the System Currency
20-3 Setting the Default Account Currency
20-3 Supporting a New Currency
20-3 Using Multiple Currencies in Your Product Offerings
20-4 Changing Currency Conversion Rates

### 21 Supporting EMU Currencies and the Euro

21-1 Supporting EMU Currencies and the Euro
21-1 Using the Euro and EMU Currencies in Your Price List
21-2 Handling Euro Conversion Rounding Errors
21-2 Rounding Errors for Overpayments and Underpayments
21-3 Changing Supported Secondary Account Currencies

### 22 Creating and Managing Customer Segments

22-1 About Customer Segments
22-2 Defining Customer Segments
# Creating and Managing Business Profiles

- About Business Profiles
- Setting Up Business Profiles and Validation Templates
- Editing the Business Profile Configuration File
  - Defining Business Profiles
  - Modifying Business Profiles
  - Deleting Business Profiles
  - Deleting Validation Templates

# Masking Sensitive Customer Data

- About Masking Sensitive Customer Data
- Masking Sensitive Data in Logs When Using Client Applications
  - Masking Additional Fields in Client Application Logs

# Validating Account Data

- About Validating Account Information
  - About ADU Account Search
  - About ADU Account Dump
    - Limiting Dump Information by Specifying a Date Range
  - About Account Data Validation
  - Configuring ADU

# Customer Management Utilities

<table>
<thead>
<tr>
<th>Utility</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>load_pin_business_profile</td>
<td>26-1</td>
</tr>
<tr>
<td>load_pin_customer_segment</td>
<td>26-3</td>
</tr>
<tr>
<td>load_pin_notify</td>
<td>26-4</td>
</tr>
<tr>
<td>load_pin_verify</td>
<td>26-5</td>
</tr>
<tr>
<td>load_transition_type</td>
<td>26-6</td>
</tr>
<tr>
<td>pin_deferred_act</td>
<td>26-7</td>
</tr>
<tr>
<td>pin_product_clear</td>
<td>26-9</td>
</tr>
<tr>
<td>pin_state_change</td>
<td>26-10</td>
</tr>
</tbody>
</table>
Preface

This guide describes how to manage customers in Oracle Communications Billing and Revenue Management (BRM).

Audience

This guide is intended for operations personnel, developers, and system administrators.

Downloading Oracle Communications Documentation

Product documentation is located on Oracle Technology Network:

http://docs.oracle.com

Additional Oracle Communications documentation is available from the Oracle software delivery Web site:

https://edelivery.oracle.com

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>E51037-01</td>
<td>December 2017</td>
<td>Initial release.</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>E51008-02</td>
<td>September 2019</td>
<td>Documentation updates for BRM 12.0 Patch Set 2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Documentation added for Customer Center.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Updated the following section:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enabling Charge Offer Purchases from Closed or Inactive Accounts</td>
</tr>
</tbody>
</table>
## List of Customer Management Features

This document lists the customer management features in Oracle Communications Billing and Revenue Management (BRM).

To learn about customer management, see *BRM Concepts*.

### Customer Management Features

This table lists customer management features.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure defaults for creating accounts. When an account is created, it is based on default settings for currency, country, and other account attributes.</td>
<td>Choosing Account Creation Options</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Collect information about customers and customer interactions</td>
<td>Making Notes about Customers</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Set up Self-Care Manager to display Web pages that your customers can use to access their accounts</td>
<td>Setting Up Customer Self Care with Self-Care Manager</td>
</tr>
<tr>
<td>Send a welcome email to a customer, or create a welcome Web page</td>
<td>Setting up Welcome Messages to Customers</td>
</tr>
<tr>
<td>Manage login names and passwords</td>
<td>Managing Passwords and Login Names</td>
</tr>
<tr>
<td>Configure options for setting account and service status</td>
<td>Managing Account and Service Status</td>
</tr>
<tr>
<td>Configure how credit card payments are handled during account creation. For example, you can charge customers when an account is created, create accounts without requiring credit-card validation, and so on.</td>
<td>Managing Credit Cards During Account Creation</td>
</tr>
<tr>
<td>Change a customer's billing configuration</td>
<td>Changing a Customer's Billing Day of Month</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Topics</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Manage a customer’s purchased charge offers</td>
<td>Canceling Charge Offers</td>
</tr>
<tr>
<td></td>
<td>Enabling Charge Offer Purchases from Closed or Inactive Accounts</td>
</tr>
<tr>
<td></td>
<td>Changing Charge Offer Quantity</td>
</tr>
<tr>
<td>Manage a customer’s purchased discount offers</td>
<td>Managing Purchased Discount Offers</td>
</tr>
<tr>
<td>Cancel discount offers</td>
<td>Canceling Discount Offers</td>
</tr>
<tr>
<td></td>
<td>Configuring Discount End Dates during Mid-Cycle</td>
</tr>
<tr>
<td></td>
<td>Rating Delayed Events for a Canceled Discount Offer</td>
</tr>
<tr>
<td></td>
<td>Changing the Status of Discounts Canceled in Mid-Cycle</td>
</tr>
<tr>
<td></td>
<td>Deleting Canceled Discount Offers</td>
</tr>
<tr>
<td>Manage a customer’s purchased bundles</td>
<td>Managing Purchased Bundles</td>
</tr>
<tr>
<td>Configure bundle and package transitions</td>
<td>Configuring Bundle and Package Transitions</td>
</tr>
<tr>
<td>Transitions determine the conditions that must be met to purchase a</td>
<td></td>
</tr>
<tr>
<td>bundle or package.</td>
<td></td>
</tr>
<tr>
<td>Backdate subscription actions; for example, cancelling or purchasing</td>
<td>Backdating Subscription Actions</td>
</tr>
<tr>
<td>a charge offer</td>
<td></td>
</tr>
<tr>
<td>Create custom service life cycles</td>
<td>Creating Custom Service Life Cycles</td>
</tr>
<tr>
<td>Custom life cycles enable you to manage service status at a detailed</td>
<td></td>
</tr>
<tr>
<td>level. For example, you can use statuses such as Dormant, Credit</td>
<td></td>
</tr>
<tr>
<td>Expired, and Fraud Investigated to indicate the exact state of a</td>
<td></td>
</tr>
<tr>
<td>service and control how the service is used.</td>
<td></td>
</tr>
<tr>
<td>Create service groups</td>
<td>Creating and Managing Service Groups</td>
</tr>
<tr>
<td>Service groups enable customers to purchase, transfer, inactivate,</td>
<td></td>
</tr>
<tr>
<td>and cancel services as a group; and to share charges, discounts,</td>
<td></td>
</tr>
<tr>
<td>promotions among a group of subscribers.</td>
<td></td>
</tr>
<tr>
<td>Create business profiles</td>
<td>Creating and Managing Business Profiles</td>
</tr>
<tr>
<td>Use business profiles to classify types of data to specify how to</td>
<td></td>
</tr>
<tr>
<td>treat the data. For example, business profiles can be used to</td>
<td></td>
</tr>
<tr>
<td>classify bills as prepaid or postpaid to determine how bills and</td>
<td></td>
</tr>
<tr>
<td>payments should be handled.</td>
<td></td>
</tr>
<tr>
<td>A profile sharing group enables an account or an account’s service</td>
<td></td>
</tr>
<tr>
<td>to share a profile with other accounts or services. A profile stores</td>
<td></td>
</tr>
<tr>
<td>ERAs or other types of information about an account.</td>
<td></td>
</tr>
<tr>
<td>Create profile sharing groups</td>
<td>Creating and Managing Profile Sharing Groups</td>
</tr>
<tr>
<td>A profile sharing group enables an account or an account’s service</td>
<td></td>
</tr>
<tr>
<td>to share a profile with other accounts or services. A profile stores</td>
<td></td>
</tr>
<tr>
<td>ERAs or other types of information about an account.</td>
<td></td>
</tr>
<tr>
<td>Manage system and account currencies</td>
<td>Managing System and Account Currencies</td>
</tr>
<tr>
<td>Customer account currencies can be different from your system</td>
<td></td>
</tr>
<tr>
<td>currency.</td>
<td></td>
</tr>
<tr>
<td>Manage Euro and non-Euro currencies</td>
<td>Supporting EMU Currencies and the Euro</td>
</tr>
<tr>
<td>Task</td>
<td>Topics</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>Create customer segments</td>
<td>Creating and Managing Customer Segments</td>
</tr>
<tr>
<td>A customer segment is a user-defined customer description that can be used to group accounts according to customer billing and payment practices, such as reliable bill payer.</td>
<td></td>
</tr>
<tr>
<td>Create business profiles</td>
<td>Creating and Managing Business Profiles</td>
</tr>
<tr>
<td>Use business profiles to classify types of data to specify how to treat the data. For example, business profiles can be used to classify bills as prepaid or postpaid to determine how bills and payments should be handled.</td>
<td></td>
</tr>
<tr>
<td>Prevent your employees from seeing sensitive customer data, such as banking information and passwords</td>
<td>Masking Sensitive Customer Data</td>
</tr>
<tr>
<td>Validate customer account data</td>
<td>Validating Account Data</td>
</tr>
<tr>
<td>You typically validate account data after certain business processes, for example, after completion of a migration or upgrade or before billing or payment allocation.</td>
<td></td>
</tr>
<tr>
<td>Run customer management utilities</td>
<td>Customer Management Utilities</td>
</tr>
</tbody>
</table>

Chapter 1

Customer Management Features
Choosing Account Creation Options

This document describes Oracle Communications Billing and Revenue Management (BRM) customer account creation options.

Topics in this document:

• About Customer Information Collected When an Account is Created
• Specifying How to Validate Customer Contact Information
• Specifying Defaults for Account Currency, Country, and Email
• Allowing Accounts to Be Created with Backdated Services or Balances
• Customizing Account Creation

See also:

• List of Customer Management Features
• BRM Concepts

About creating customer accounts

Account creation typically follows this process:

1. A customer contacts one of your customer representatives.

About Customer Information Collected When an Account is Created

When you create an account, you collect information about the customer such as:

• The package that the customer signs up for.
• Contact information, such as name, address, and telephone number.
• Service login and Web access names and passwords.
• Customer’s language.
• Customer’s account currency.
• Billing information, such as:
  – The accounting type (balance forward or open item).
  – The frequency of the billing cycle.
  – The payment method (credit card, invoice, and so on).
  – Payment information, such as credit card number or invoice billing address.

The information required by default is the last name, address, city, and country. You can change which information is required by using the Field Validation Editor.
If you create accounts by using a custom application or over the Web, you can create your own account creation fields. To do so, you need to create new fields in Storable Class Editor. The information you collect is stored in account profiles.

**Specifying How to Validate Customer Contact Information**

You can define the valid formats for your customers’ contact information by using the Field Validation Editor. For example, you can require phone numbers to contain 10 digits and a dash (-) character.

You can also specify whether the information is required, whether it’s case sensitive, and the minimum and maximum values (for example, the maximum length of a login name).

You can configure this information:

- **Login name format.** The default requirement is at least 1 character but no more than 255.
- **Password format.** The default requirement is at least 1 character but no more than 255.
- **Telephone number format.** BRM includes several standard formats, but you can add your own formats.
- **US state abbreviation.** The default required format is two uppercase letters such as CA.
- **Salutation.** For example, Mr. or Ms.
- **US ZIP code format.** Tax calculation programs validate ZIP codes. If you use a tax calculation package, make sure that your tax calculation program can handle 9-digit ZIP codes.

**Specifying Defaults for Account Currency, Country, and Email**

You can set up these system-wide defaults that are used by all account creation methods:

- Default account currency
- Default country
- Default customer email domain name

To specify these defaults:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Edit these entries:
   - To specify the default account currency, enter a currency code in the `currency` entry.
     For a list of currency codes, see `BRM_home/include/pin_currency.h`.
   - To specify the default country, change the value of the `country` entry:
     ```
     fm_cust_pol     country     USA
     ```
The country name must conform to the validation rules for country names set in the Field Validation Editor. See "Specifying How to Validate Customer Contact Information ".

- To specify the default email domain, change the value of the domain entry.
  - fm_cust_pol domain isp.nz

3. Save the file.
The new values become effective immediately and apply to the next account created. You don't need to restart the CM.

Allowing Accounts to Be Created with Backdated Services or Balances

By default, when you create an account, its associated services and balances must have a creation date on or after the account creation date. You can configure BRM to allow users to create accounts with the service or balance date backdated prior to the account creation date.

To enable this feature, run the pin_bus_params utility to change the SubsDisableBackdatedValidations business parameter. For information about this utility, see BRM Developer's Guide.

To allow accounts to be created with backdated services or balances:

1. Go to BRM_home/sys/data/config.
2. Create an XML file from the /config/business_params object:

   pin_bus_params -r BusParamsSubscription bus_params_subscription.xml

3. In the file, change disabled to enabled:

   <SubsDisableBackdatedValidations>enabled</SubsDisableBackdatedValidations>

4. Save the file as bus_params_subscription.xml.
5. Load the XML file into the BRM database:

   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 Account Creation

See BRM Opcode Guide for information about the following customizations:

- To customize how account names are displayed, use the PCM_OP_CUST_POL_PREP_ACCTINFO policy opcode. For example, you can remove spaces from the name or capitalize the first and last name. The default is to store the name as it was entered at account creation.
- To change the format of account numbers, customize the PCM_OP_CUST_POL_PREP_ACCTINFO policy opcode.
By default, the account number is derived automatically from the account object POID. For example, if the POID is:

0.0.0.1 /account 12225 19

The account number is:

0.0.0.1-12225

The account number does not need to include any part of the POID; you can define your own format.

- To change the date format, customize the PCM_OP_CUST_POL_PREP_PAYINFO policy opcode. By default, BRM stores expiration dates as four-character strings in the format MMYY. The years from 00 and beyond are understood as the 2000 millennium. Valid formats are MMYY, MM/YY, M/YY, and MM/YYYY, all of which are converted to MMYY.

- To check for duplicate account creations, use the customer policy opcodes. For example, you can customize your customer validation policies to check for a duplicate name or credit card number. This allows you to prevent customers from accidentally signing up and getting billed for accounts they do not want. This might happen when a customer loses a connection right after creating an account. Being unsure if the account was created, the customer creates a duplicate account.

- To validate account creation numbers, customize the PCM_OP_CUST_POL_VALID_AACINFO policy opcode. When creating accounts, you can offer different packages to different customers by having customers enter account creation numbers. You can also assign account creation numbers to customers to track marketing information.

- To return a point-of-presence (POP) list, customize the PCM_OP_CUST_POL_GET_POPLIST policy opcode.

- To prepare automatic account creation (AAC) data for validation, use the PCM_OP_CUST_POL_PREP_AACINFO policy opcode.

- To export information to a non-BRM database, customize the PCM_OP_CUST_POL_POST_COMMIT policy opcode.

- To customize how to select schemas during account creation in a multischema system, customize the PCM_OP_CUST_POL_GET_DB_LIST and PCM_OP_CUST_POL_GET_DB_NO opcodes.

- To store information collected when customers create an account by using a Web site, customize the PCM_OP_CUST_POL_PREP_AACINFO policy opcode.
Collecting Information About Customers

This document describes features you can use to collect and manage information about customers in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

• Making Notes about Customers
• Collecting a Summary of Activities Performed on an Account
• Customizing the List of Reasons for Account Changes
• Maintaining an Audit Trail of BRM Activity

See also:

• List of Customer Management Features
• BRM Concepts

Making Notes about Customers

You can add notes to accounts to record information obtained from customers and to explain why you performed various actions.

Both Billing Care and Customer Center categorize notes to differentiate them when they are displayed in the Notes dialog box and to enable other tools, such as reporting applications, to gather a specific type of note from your BRM database. Note categories include:

• General Notes
• A/R Charge Credit Card
• A/R Credit Account
• A/R Credit Item
• A/R Debit Account
• A/R Open Dispute
• A/R Set Credit Limit
• A/R Settle Dispute
• A/R Write-Off
• Account/Service Status Change

General notes, such as notes about the best time of day to call a customer, apply to the entire account. Notes in all other categories apply to specific actions, such as adjustments, charges, credits, disputes, write-offs, and status changes.

A CSR can retrieve a list of notes associated with an account. Each note includes the name of the CSR who created.
Collecting a Summary of Activities Performed on an Account

You collect a summary of activities performed on an account by using New Feed. Enable News Feed for each event you want to collect data about.

You can enable News Feed for the following events:

- **Accounts receivable (A/R)**, which includes currency adjustments, noncurrency adjustments, open disputes, closed disputes, refunds, write-offs, write-off reversals, and collections.
- **Payments**, which includes payments, payment method changes, and payment reversals.
- **Account**, which includes name or contact information changes, account status changes, service status changes, and services (for example, a SIM card) attached or detached from the account, and billing information changes.
- **Charges**, which includes one-time charges, recurring charges, cycle charges, purchase charges, purchase and cancellation of a package or bundle, bill issued, and bill issued mid-cycle.

To enable News Feed for an event:

1. Open the `BRM_home/sys/data/config/pin_notify` file in a text editor.
2. Uncomment the entry for the events for which you want to enable News Feed:
   
   ```
   180  0    event_name
   ```

   where `event_name` is the name of the event for which you want a summary of.

   For example:
   
   ```
   180  0    /event/billing/account/adjustment
   ```

   This example means that opcode 180 (PCM_OP_ACT_PROCESS_EVENTS) is called when the `/event/billing/account/adjustment` event occurs.

3. Save and close the file.
4. Run the following command, which loads the event notification file into the BRM database:
   
   ```
   load_pin_notify  pin_notify_file
   ```

5. Stop and restart the Connection Manager (CM).

Customizing and Localizing News Feed

The strings for News Feed contain the basic text of a message and include the type of activity performed, such as account, dispute, payment and so on.

You can customize and localize the strings for News Feed; for example “Payment reversed”. To do so, you edit a copy of the `newsfeed.en_US` sample file in the `BRM_home/sys/msgs/newsfeed` directory. If you are loading a localized version of this file, use the correct file extension for your locale. You then use the `load_localized_strings` utility to load the contents of the file into the `/strings` objects.

When you run the `load_localized_strings` utility, use this command:
To load the customized strings for News Feed:

1. Open the `BRM_home/sysmsgs/newsfeed/newsfeed.locale` file in a text editor.
2. If you want to add new strings for News Feed, add the strings and map them to the reason codes from a list of reasons appropriate to the event type that are defined in the `newsfeed.locale` configuration file.
   For example:
   
   ```
   DOMAIN = "newsfeed"
   STR
   ID = 132 ;
   VERSION = 60 ;
   STRING = "Payment reversed" ;
   END
   STR
   ID = 133 ;
   VERSION = 60 ;
   STRING = "%s original payment, %s" ;
   END
   ```
3. If you want to modify the existing strings for News Feed, search for the string and change the string value.
4. Save and close the file.
5. Load the strings using the `load_localized_strings` utility.

**Customizing the List of Reasons for Account Changes**

When making changes to accounts in Billing Care or Customer Center, CSRs can choose from a list of reasons that specify why a change was made.

By default, BRM provides reasons for changing an account such:

- Activating or inactivating an account
- Closing or writing off an account
- Crediting, debiting, or adjusting an account
- Updating the payment method
- Updating tax information
- Charging an account

You can add, modify, or delete reasons by editing the `reasons.en_US` sample file in the `BRM_home/sysmsgs/reasoncodes` directory, where `BRM_home` is the directory
in which the BRM server software is installed. You then use the
load_localized_strings utility to load the contents of the file into the /strings objects.

To run the load_localized_strings utility, use the following command:

load_localized_strings reasons.locale

where locale is the file extension (for example, en_US). If you are loading a localized
version of this file, use the correct file extension for your locale.

For more information about the load_localized_strings utility, see BRM Developer’s
Guide.

Maintaining an Audit Trail of BRM Activity

BRM provides support for auditing any object in the BRM database so that a record is
kept of every version of the object for future reference. This can be used to track
changes to customer profiles, customer payment information, and so on. An audit trail
can also be used to track internal changes, such as changes to your price list.

Note:

Enabling an audit trail decreases system performance significantly. Keep an
audit trail only for the BRM activity you feel is absolutely necessary.

To access audit trail information for a customer, you must write an application. Your
application would take the pertinent customer account information (such as the
customer account number and the general time the audit occurred) and find and
retrieve the requested audit trail data.

See "Auditing Customer Data" in BRM Developer’s Guide for the following:

• Instructions on accessing audit trail information from the BRM database.
• Information on how audit trails work.
• Instructions on making new and existing object fields auditable.
Setting Up Customer Self Care with Self-Care Manager

This chapter describes how to set up Oracle Communications Billing and Revenue Management (BRM) Self-Care Manager to display Web pages that your customers can use to access their accounts. It includes a procedure for configuring the application server.

To add or change Self-Care Manager functionality, see “Customizing the Self-Care Manager Interface” in BRM Developer’s Guide.

See also:

• List of Customer Management Features
• BRM Concepts

About Self-Care Manager

Self-Care Manager includes the following components:

• A set of Java Server Pages (JSPs) that display HTML forms in a standard Web browser. The JSPs can display data, such as a list of available packages, from the BRM database.

  You can customize the JSPs to change the appearance of the Web pages or to change the data that you receive from or display to customers. For more information about JSPs, see Sun Microsystems’ Java documentation.

• A set of BRM Business Application SDK (BAS) beans that provides an interface between the JSPs and BRM. The BAS beans connect to a Connection Manager (CM) and transfer data to and from BRM.

  A programmer can create custom controllers and other Java components to customize the data that you display or receive from customers. For more information, see “Customizing the Self-Care Manager Interface” in BRM Developer’s Guide.

• An application server that processes requests originating from customer Web browsers, manages customer sessions, and transfers data to and from the BAS beans. For more information, see the JavaSoft documentation.

The JSPs and the application server are run by your Web server.

Supported Application Servers

Self-Care Manager supports the Tomcat and WebLogic application servers. Figure 4-1 shows the flow of information.
Figure 4-1    Self-Care Manager System Architecture

For installation instructions, see "Installing Self-Care Manager" in *BRM Installation Guide*.

About Customizing Self-Care Manager

Self-Care Manager can be customized as follows:

- You can change the appearance of Self-Care Manager pages by editing the HTML files, JSPs, and style sheet. This requires HTML knowledge. See "Changing Web pages" for more information.
- You can modify or add source code to collect additional information from your customers or to provide your customers with additional options. This requires the Customer Center SDK as well as Java programming knowledge.

About Currencies

All accounts created in Self-Care Manager use the default account currency for your BRM system. Your customers do not have the option of selecting a currency when they create an account through Self-Care Manager.

You can change the default currency Self-Care Manager uses by changing the default account currency. See "Setting the Default Account Currency".

If you want customers to have the option of selecting a currency, a programmer can customize Self-Care Manager to add a currency field. See "Customizing the Self-Care Manager Interface" in *BRM Developer's Guide*.

Configuring the Application Server

You can deploy your Self-Care Manager application with these application servers:

- Apache Tomcat. See "Setting up Apache Tomcat".
- WebLogic. See "Setting Up Oracle WebLogic".
Setting up Apache Tomcat

This section explains how to deploy your Self-Care Manager application with Apache Tomcat.

**Note:**

Self-Care Manager is not compatible with Tomcat in a clustered environment. Self-Care Manager should not have the `<distributable/>` element set in the `web.xml` file.

Requirements

- Your Self-Care Manager **WAR** file. See "Customizing the Self-Care Manager Interface" in *BRM Developer’s Guide*.
- The directory in which Self-Care Manager is installed. The default is `BRM_home/WebKit/WebKit`, where `BRM_home` is the directory in which the BRM server software is installed.

Deploying Your Application with Apache Tomcat

To deploy your Self-Care Manager application with Apache Tomcat:

1. Install Apache Tomcat. For information, see the Apache Tomcat documentation.
2. Deploy your **WAR** file as a new Web component by using the Apache Tomcat interface.
3. Copy the `WebKit.properties` and `Infranet.properties` files from `BRM_home/WebKit/WebKit` to `Tomcat_home/lib`.
4. Stop and restart the Tomcat server.

Your Self-Care Manager application is now deployed.

Setting Up Oracle WebLogic

This section explains how to deploy your Self-Care Manager application with Oracle WebLogic.

Requirements

- Creating a new WebLogic domain.
- Your Self-Care Manager **WAR** file. See "Customizing the Self-Care Manager Interface" in *BRM Developer’s Guide*.
- The default directory in which Self-Care Manager is installed. The default is `C:\Program Files\Portal Software\Self Care Manager`.
Deploying Your Application with Oracle WebLogic

To deploy your Self-Care Manager application with WebLogic:

1. Install WebLogic. For information, see the WebLogic documentation.
2. Create WebLogic domain. For information, see the WebLogic documentation.
3. Copy the `WebKit.properties` and `Infranet.properties` files from `C:\Program Files\Portal Software\Self Care Manager` to `WebLogic_home\mydomain\applications\webkit_en\WEB-INF\classes`.
4. Start the WebLogic server.
5. Deploy your WAR file as a new Web component by using the WebLogic interface. For information, see the WebLogic documentation.
6. Stop and restart the WebLogic server.

Your Self-Care Manager application is now deployed.

Deploying Self-Care Manager

Your Self-Care Manager installation includes the user interface files (JSPs, HTML files, and GIFs) for the standard browser version of Self-Care Manager. The files are in the `htmlui_en` directory.

**Note:**

This is the directory name for the English locale. Directory names for localized versions use the appropriate locale in place of `en`.

One approach to deploying Self-Care Manager so your subscribers can access it:

- Configure your Web server to look for `index.html` as the default HTML home page.
- Have subscribers enter `http://hostname\WebKit/htmlui_en` in the Web browser.

**Note:**

If your Web server does not support specifying a default home page for each application, subscribers will need to enter the full URL, including the file name for the home page. For example, `http://hostname\WebKit/htmlui_en/index.html`.

Supporting Localized Versions of Self-Care Manager

If you have customers in different countries, you can set up localized versions of the Self-Care Manager pages. You obtain localized versions of Self-Care Manager from
BRM or use the Localization SDK to create localized versions. In both cases, you install a Self-Care Manager WAR file just as you do for English.

You can deploy multiple localized versions of Self-Care Manager in one of these ways:

- **Set up a separate application server for each locale.** You set up multiple application servers with a single Web server. For information, see your application server documentation.

  You can then follow the standard procedure for configuring your application server for each combination of server and locale. For each server, deploy the WAR file for a different locale.

- **Set up a separate Web application in your application server for each locale.**

  **Note:**

  If you use this method, you will encounter problems later if you try to use opcode and flist logging for troubleshooting.

  For information on creating localized versions of Self-Care Manager, see "Localizing Self-Care Manager" in *BRM Developer's Guide*.

### Using Self-Care Manager

This section describes how to set up Self-Care Manager for use.

### Setting Connection Parameters

Self-Care Manager uses information in the `WebKit.properties` file to establish a connection to BRM. This properties file includes the standard BRM connection parameters, such as database number, login type, and password.

**Note:**

BRM connection parameters are also in the `Infranet.properties` file.

To set connection parameters:

1. Open the Self-Care Manager properties file (`SelfCareManager_install_dir/WebKit.properties`).
2. Edit these entries:
   ```
   user=root.0.0.0.1
   password=password
   host=hostname
   port=11960
   ```
3. Save and close the file.
4. Open the `Infranet.properties` file (`SelfCareManager_install_dir/Infranet.properties`).
5. Edit the `infranet.connection` entry; for example:

```text
infranet.connection=pcp://root.0.0.0.1:password@host:40010/service/pcm_client 1
```

In this example:
- The login name is `root.0.0.0.1`.
- The password is `password`.
- The hostname is `host`.
- The port is `40010`.

6. Save and close the file.

---

**Note:**
To improve security, change the default root login name (root.0.0.0.1) and the default password (password). See “Configuring Login Names and Passwords for BRM Access” in BRM System Administrator’s Guide.

---

### Specifying which Package List to Display

By default, Self-Care Manager displays packages included in the `webclient` package list. If the `webclient` package list does not exist, Self-Care Manager displays packages included in the `default-new` package list.

To specify a package list for Self-Care Manager:

1. Open the Self-Care Manager properties file (`SelfCareManager_install_dir/WebKit.properties`).
2. Enter the name of the package list in the `Infranet.PricingPlan` entry; for example:
   ```text
   pricingplan=webclient
   ```
3. Save the file.

### Setting the Timeout

You can specify the length of time before a timeout when there is no customer activity.

You change the timeout setting in your application server.

### Enabling a Single Login for Multiple Services

By default, Self-Care Manager requires your customers to enter a separate login and password for each service when they purchase bundles with more than one service.

You can set up Self-Care Manager to require only a single login and password that applies to all services. To do this, edit the Self-Care Manager properties file as follows:

1. Open the Self-Care Manager properties file (`SelfCareManager_install_dir/WebKit.properties`).
2. Change the `singleLogin` entry to:
   ```text
   singleLogin=true
   ```
3. Save and close the file.

Optimizing Self-Care Manager Connection Pool Performance

Self-Care Manager uses connection pooling. When a BRM connection is required for an opcode call, the connection is made for a brief period of time, instead of for the entire duration of a user session, and is then released back to the connection pool. This enables each BRM connection to handle multiple customer connections.

To optimize Self-Care Manager performance, you can change the following connection pooling parameters in the Self-Care Manager properties file (SelfCareManager_install_dir/WebKit.properties):

- **infranet.bas.connectionpool.size**: The maximum number of connections in the connection pool. More connections can improve performance but put a heavier load on the BRM server. Any value greater than 0 is valid. The default is 4.
  
infranet.bas.connectionpool.size = 4

- **infranet.bas.connectionpool.timeout**: The maximum time in milliseconds that customers wait for a BRM connection before getting notified that a connection is not available. If customers do not get a connection within this time, they get an “error communicating with BRM” message.

  You must add this parameter to the Self-Care Manager properties file (WebKit.properties) using a value appropriate for your installation. Any value greater than 0 is valid.

  infranet.bas.connectionpool.timeout = connection_timeout_in_milliseconds

  The default is 30000 (30 seconds).

Adding a Service to the Self-Care Manager Home Page

The Self-Care Manager home page includes a list of services available to your customers. If you add an optional manager to your BRM system, and that optional manager uses Self-Care Manager, you need to add those service types to the list in the home page.

**Note:**

If you have multiple Self-Care Manager home pages, you need to add those service types to each home page.

To add a service:

1. Open the Self-Care Manager home page with an HTML editor or text editor. The default Self-Care Manager home page is index.html.

2. Locate this section of the file:

   `<SELECT Name="service" Size="1">
     <OPTION Value="ip" SELECTED>ip
     <OPTION Value="email">email
     <OPTION Value="telephony">telephony
   </SELECT>`
3. For each service you want to appear on the list, add a new option to this list. For **Value**, enter the BRM name of the service. The name outside the **OPTION** tag is the name displayed on the Web page.

For example:

- `<OPTION Value="gsm/sms">GSM SMS</OPTION>`
- `<OPTION Value="gsm/telephony">GSM Telephony</OPTION>`
- `<OPTION Value="gsm/data">GSM Data</OPTION>`

4. Save and close the file.

---

**Troubleshooting Self-Care Manager**

This section contains troubleshooting information about these Self-Care Manager issues:

- The error logs show an “error communicating with Portal” message
- The Web pages display errors

### The error logs show an “error communicating with Portal” message

To debug this type of error, you turn on opcode and flist logging for the Java PCM. This logs the input and output flists for each opcode that Self-Care Manager calls. You can use these flists to see problems communicating with BRM.

To turn on opcode and flist logging:

1. Open the **Infranet.properties** file in a text editor. The file is located in your Self-Care Manager installation directory.

2. Add these lines to the file:

   ```properties
   infranet.log.opcodes.enabled=true
   infranet.log.opcodes.file=pathname/opcodes.log
   ```

   where `pathname` is the path to the directory for the log file. For example:

   ```properties
   infranet.log.opcodes.file=d:/temp/opcodes.log
   ```

   **Note:**
   You must use forward slashes in the directory paths.

3. Stop and restart the application server.

4. Save and close the **file**.

5. Reproduce the error and check the file `applicationserver_install_dir/servers/default/javaPCM.log`.

### The Web pages display errors

The Web pages might display an error message or display “Click to continue” in place of the correct page. To debug this type of problem, you enable debugging in the **error.jsp** file so that exceptions are written to the log file `debug_errorlog.jsp`. This prints a stack trace, which can help developers locate where an exception occurred.
To enable debugging in the error.jsp file:

1. Open the appropriate version of the error.jsp, located in the htmui_en directory in the Self-Care Manager installation directory.
2. Uncomment this line by removing < and >:
   ```html
   <%@ include file = "debug_errorlog.jsp" %>
   ```
3. Save and close the file.

Using the Default HTML Web Pages

Self-Care Manager supplies a set of JSPs that you can use as part of your customer self-care Web site. These JSPs serve HTML pages.

This section describes Self-Care Manager's default HTML pages and provides an overview of the functionality they offer.

For information on customizing these pages, see "Changing Web pages" and "Customizing the Self-Care Manager Interface" in BRM Developer's Guide.

Logging In

Figure 4-2 shows the login page your customers see when they access your self-care Web site:

Figure 4-2 Login Page for Self-Care Web Site

Self-Care Manager supports email, IP, and telephony service types.

Accessing Account Information

After customers log in to your Web site on the login page, the Account Information page appears. This page displays the details of the account bill unit (billinfo object)
and balance groups associated with the bill unit, and the current balance of all the balance groups.

To view the details of the other bill units in the account, customers click the Other Bill Units Information link on the Account Information page and select the bill unit. The details will be displayed in the Account Information page. For information, see "Accessing Account Information".

From this page, customers can perform the following tasks:

- Finding or Searching for and Viewing Events
- Applying Voucher Top-Ups
- Viewing Resource Reservation Details
- Viewing Invoices
- Viewing Account Activity
- Viewing Products or Offers Purchased
- Paying Bills Online
- Viewing Service Details for a Bill Unit
- Viewing Bill Units in an Account
- Changing Login Name, Password, or Account Status

Finding or Searching for and Viewing Events

Customers can specify the search criteria and view the events of the selected bill unit. To enable this feature, you must configure the item types in the WebKit.properties file.

To specify the search criteria, customers click the Event Search link on the Account Information page. The Event Search page appears. Customers then specify the search criteria and click Search.

To view the events of other bill units in the account, see "Viewing Bill Units in an Account".

Based on the search criteria and the item types configured in WebKit.properties, events are retrieved and displayed in the Event Search page.

For example:

- To view all the events related to cycle_forward, add /cycle_forward to the key EventSearch.ItemTypes in WebKit.properties:
  
  EventSearch.ItemTypes=/cycle_forward

- To view all the events related to adjustment and payment, add /adjustment and /payment to the key EventSearch.ItemTypes in WebKit.properties:

  EventSearch.ItemTypes=/adjustment,/payment

Applying Voucher Top-Ups

Customers use vouchers to top-up their currency and noncurrency account balances. They can top-up one currency balance and an unlimited number of noncurrency
balances. For more information, see "About Standard Top-Ups" in BRM Configuring and Collecting Payments.

To top-up their account with a voucher, customers click the Voucher Top-Up link on the Account Information page. The Voucher Top-Up page appears. Customers can then enter the voucher ID and pin and click Validate to check the balances and validity of the specified voucher. If the voucher is valid, the Apply button is enabled.

If there are multiple balances in the bill unit, customers can select the balance groups whose account balance they want to top up.

To apply the vouchers immediately and top-up the account balances, customers select the required balance groups, select Allocate Now, and click Apply.

Customers can also choose just to apply the voucher to the balance groups and not top up the account balance. To do this, the customer clicks Apply without selecting Allocate Now. The voucher can later be allocated but only through Customer Center.

Viewing Resource Reservation Details

Customers can view the resource reservation details of a bill unit. To do this, customers click the Resource Reservation link on the Account Information page. It displays available credit limit, total bill amount, amount reserved, and available balance of all the balance groups in a bill unit.

To view the reservation details, customers click the Reservations link and the details are displayed in the Resource Reservations Details page.

If the credit limit of any of the balance groups is unlimited, the Credit Limit value is displayed as Unlimited.

To find the balance group, which has unlimited credit balance customers, click the Unlimited link displayed next to the Credit Limit. The Included Services page appears. See "Viewing Service Details for a Bill Unit".

Viewing Invoices

Customers can view their invoice details by generating a report of their past bills. To view the invoice details, customers click the Invoice link on the Account Information page. The Invoice Selection page.

Customers can then select new start and end dates for the report or keep the default dates that represent the last billing cycle. After clicking Submit, the Invoice page appears, which lists invoices available for the selected period. In this page, customers click the invoice they want to view.

Viewing Account Activity

To view their account details, customers can generate a report. To generate a report, customers click the Account Activity link on the Account Information page.

Customers can then select new start and end dates for the report or keep the default dates which represent the last billing cycle.

After clicking Submit, the Account Activity page appears which shows the account's details; for example, customer's name, address, and event details in chronological order.
If the customer uses additional services, some types of usage activity appear on separate pages. Links for the additional pages appear on the line just above the list of events.

Viewing Products or Offers Purchased

To view the products or offers purchased, customers click the Products link on the Account Information page. The Product Information page.

If a discount was purchased as part of a bundle, an icon appears before the bundle name.

Paying Bills Online

To pay bills online, customers click the Online Payment link on the Account Information page.

To pay the bill, customers enter an amount and credit card information and click Submit.

BRM updates the customer's balance the next time you run billing and collects the BRM-initiated payment.

For more information, see "About Collecting BRM-Initiated Payments" in BRM Configuring and Collecting Payments.

To view online bill payment records, customers click the Online Payment Audit link on the Online Payment page. A page appears on which they enter start and end dates. Self-Care Manager displays the payments made between the start and end dates.

Viewing Service Details for a Bill Unit

To view the list of services and the details associated with all the balance groups in a bill unit, customers click the Included Services link on the Account Information page. The Service Details page appears and displays the details of the services.

Viewing Bill Units in an Account

To view all the bill units in an account, customers click the Other Bill Units Information link on the Account Information page. The Service Level Bill Unit Information page appears and displays all the bill units.

To view the details of a specific bill unit, click on the bill unit. The details are displayed in the Account Information page. For information, see "Accessing Account Information".

Changing Login Name, Password, or Account Status

Customers can change their login name, password, or account status by clicking the appropriate link on the Account Information page.

If customers click Change a Login or Change Status, the Change Service Login page appears.

The account status can be active, inactive, or closed.
Changing Web pages

This section describes editing Self-Care Manager files to change the appearance of the pages and to make other modifications. You can make these types of changes if you are familiar with HTML tags and JSP tags. For information on making more extensive customizations to Self-Care Manager, see “Customizing the Self-Care Manager Interface” in *BRM Developer's Guide*.

Editing Self-Care Manager Files

To change the appearance of Web pages, edit the following user interface files, located by default in the `htmlui_en` directory:

- **JSPs**: The files containing the text and forms that provide the user interface on the pages. JSPs are HTML pages with added JSP tags. You can use any text editor and some HTML editors to edit the JSPs. For information on modifying these pages, see "Editing JSPs".

- **HTML pages**: The HTML account creation page and files with common HTML code used by the JSPs. For example, the `header.html` file includes the HTML `HEAD` tag. You can use a text editor or HTML editor to edit these files.

- **Cascading style sheet (CSS)**: The file that defines the appearance of the HTML pages. Edit this file to make global formatting changes, including fonts, text alignment, and margins. You can use a text editor or a CSS editor to edit this file.

- **GIF files**: The graphics displayed in the HTML pages. You can edit the files with a graphics program or replace them with your own graphics.

**Note:**

Before editing, make a copy of the default pages, including all HTML pages and JSPs, and edit and test the copies.

When you edit a Self-Care Manager file, you should edit the version located in the Self-Care Manager *WAR* file. This ensures that the customized files are used if you reconfigure your servlet engine for Self-Care Manager in the future. See "Modifying the Self-Care Manager WAR File".

You can also use the Customer Center SDK to modify Self-Care Manager files from the *WAR* file. See *BRM Developer's Guide*.

Editing JSPs

JSPs use standard HTML formatting. In addition, the JSPs use special code to display information, or to display a text entry field for customer input. Each JSP tag starts with `<%` and ends with `%>`.

You can move data input and display codes from one location on the page to another. You can also delete them. For example, if you do not offer the IP telephony service, you can delete all code that displays IP telephony information.
Note:

- When editing JSPs, be careful to maintain the syntax when moving or deleting code. For example, a missing `%>` from a JSP tag will cause problems.
- When removing code, do not remove input entry points for information that is required for account creation. See "Customizing the Self-Care Manager Interface" in BRM Developer's Guide.
- Do not copy data input and display codes from one page to another. Each code works only on the page that includes it.

For more information about editing JSPs, see Sun Microsystems' JSP documentation or other JSP reference documents. For more extensive JSP customization, see "Customizing the Self-Care Manager Interface" in BRM Developer's Guide.

Modifying the Self-Care Manager WAR File

When you edit Self-Care Manager files, you should edit the versions in the Self-Care Manager Web Application Archive (WAR) file. This file is created when you install Self-Care Manager, and it is the source for the Self-Care Manager files that the servlet engine copies when you configure it to run Self-Care Manager.

If you configured your servlet engine before modifying individual JSPs or other files, you also must copy your modified files to the servlet engine directory where Self-Care Manager files are located.

By modifying the files in the WAR file, you make sure that you do not overwrite your modified files if you reconfigure your servlet engine for Self-Care Manager in the future.

Following are instructions for modifying the Self-Care Manager WAR file. Alternatively, you can use the Customer Center SDK to create a customized WAR file. See BRM Developer's Guide.

To modify the Self-Care Manager WAR file, you need to install the Java Development Kit (JDK) on your system. The system running your application server already has the JDK. For information on supported versions, see "BRM Software Compatibility" in BRM Developer's Guide.

To modify the Self-Care Manager WAR file:

1. Copy the SelfCareManager_install_dir/WebKit/webkit_en.war file to a temporary directory.
2. Extract the files from the WAR file using the jar command, as follows:
   ```sh
   jar xvf webkit_en.war
   ```
3. Delete or move the copy of webkit_en.war in the temporary directory.
4. Edit any files in the htmlui_en directory.
5. Create a new WAR file using the jar command from the temporary directory where you extracted the original files:
   ```sh
   jar cvf webkit_en.war
   ```
6. Copy this **WAR** file to *SelfCareManager_install_dir/WebKit*.

When you set up Self-Care Manager as an application in your servlet engine, it will include the files you modified.

If you have already set up Self-Care Manager in the servlet engine, copy the modified files to the correct location in the servlet engine's directory.

**Files Used by Self-Care Manager**

This section describes the various types of files used by Self-Care Manager.

**JSPs Used by Self-Care Manager**

*Table 4-1* describes the JSPs used by Self-Care Manager:

<table>
<thead>
<tr>
<th>Description</th>
<th>File name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enables customers to change the login name for a service.</td>
<td>change_login_form.jsp</td>
</tr>
<tr>
<td>Enables customers to change a password.</td>
<td>change_passwd_form.jsp</td>
</tr>
<tr>
<td>Enables customers to change account status.</td>
<td>change_status_form.jsp</td>
</tr>
<tr>
<td>Defines strings for constants commonly used in other Self-Care Manager files.</td>
<td>constants.jsp</td>
</tr>
<tr>
<td>Enables error logging for debugging.</td>
<td>debug_errorlog.jsp</td>
</tr>
<tr>
<td>Handles errors and displays error messages for all JSPs.</td>
<td>error.jsp</td>
</tr>
<tr>
<td>Contains standard information displayed at the end of all JSPs.</td>
<td>footer.jsp</td>
</tr>
<tr>
<td>Enables customers to select the start time and end time used for generating the invoice.</td>
<td>invoice_selection.jsp</td>
</tr>
<tr>
<td>Used by other JSPs to retrieve commonly used account data saved during the session, such as the name and address for the account.</td>
<td>load_session.jsp</td>
</tr>
<tr>
<td>Verifies the login name and password and displays confirmation when a customer logs in.</td>
<td>login_verify.jsp</td>
</tr>
<tr>
<td>Displays confirmation that a customer logged out and ends the session.</td>
<td>logout.jsp</td>
</tr>
<tr>
<td>Enables customers to pay an outstanding bill online with a credit card.</td>
<td>online_payment.jsp</td>
</tr>
<tr>
<td>Enables customers to search for records of online bill payments. The customer specifies a start date and end date.</td>
<td>online_payment_audit.jsp</td>
</tr>
<tr>
<td>Enables customers to purchase additional packages.</td>
<td>purchase_plan_form.jsp</td>
</tr>
<tr>
<td>Enables customers to select the day, month, and year when specifying a time range for viewing account activity or invoices.</td>
<td>selection.jsp</td>
</tr>
<tr>
<td>Saves the current user's session ID.</td>
<td>session_id.jsp</td>
</tr>
</tbody>
</table>
Table 4-1  (Cont.) Self-Care Manager JSPs

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>show_invoice.jsp</td>
<td>Displays a customer invoice.</td>
</tr>
<tr>
<td>usage_report.jsp</td>
<td>Displays details about customer activity, such as a list of IP telephony calls. HTML pages use UsageReportGeneral.jsp instead.</td>
</tr>
<tr>
<td>usage_selection.jsp</td>
<td>Enables customers to set the start and end dates for a usage report. For example, a customer can specify to see IP telephony calls for more than a single month.</td>
</tr>
<tr>
<td>UsageReportCommon.jsp</td>
<td>Included in the other usage report JSPs to do event search tasks that are common to all the usage report JSPs.</td>
</tr>
<tr>
<td>UsageReportContentView.jsp</td>
<td>Displays details about a customer's content usage.</td>
</tr>
<tr>
<td>UsageReportGeneral.jsp</td>
<td>Displays details about general customer activity, including cycle forward fees, purchase fees, adjustments, and IP and email usage. Other usage report JSPs display activity for specific types of usage.</td>
</tr>
<tr>
<td>UsageReportGprs.jsp</td>
<td>Displays details about a customer's mobile data (GPRS) usage.</td>
</tr>
<tr>
<td>UsageReportGsm.jsp</td>
<td>Displays details about a customer's mobile voice (GSM) usage.</td>
</tr>
<tr>
<td>UsageReportSms.jsp</td>
<td>Displays details about a customer's mobile messaging (SMS) usage.</td>
</tr>
<tr>
<td>UsageReportView.jsp</td>
<td>Included in the other usage report JSPs to display the data from a usage report on the Web page.</td>
</tr>
<tr>
<td>view_balance.jsp</td>
<td>Displays the customer's account balance summary and has links to resource or balance reservation, event search, and voucher top-up pages.</td>
</tr>
<tr>
<td>view_invoice.jsp</td>
<td>Enables customers to specify which invoice to display.</td>
</tr>
<tr>
<td>view_product.jsp</td>
<td>Displays information about a customer's purchased products or charge offers.</td>
</tr>
<tr>
<td>AccountBalancePrepaid.jsp</td>
<td>Display the resource or balance reservation details of a bill unit.</td>
</tr>
<tr>
<td>AccountBalancePrepaidView.jsp</td>
<td></td>
</tr>
<tr>
<td>ReservationInfo.jsp</td>
<td></td>
</tr>
<tr>
<td>ReservationInfoView.jsp</td>
<td></td>
</tr>
<tr>
<td>Event_details.jsp</td>
<td>Enable customers to view the events of the selected bill unit.</td>
</tr>
<tr>
<td>Event_search.jsp</td>
<td></td>
</tr>
<tr>
<td>eventssearch_dateselection.jsp</td>
<td></td>
</tr>
<tr>
<td>Voucher_details.jsp</td>
<td>Enables customers to top-up account balances by using a voucher.</td>
</tr>
</tbody>
</table>

HTML Pages Used by Self-Care Manager

Table 4-2 shows the HTML files used by Self-Care Manager:

Table 4-2  HTML Files Used by Self-Care Manager

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>footer.html</td>
<td>HTML code that you can use as a footer on every JSP page by including this file name in the JSP. For example: &lt;jsp:include page=&quot;footer.html&quot;/&gt;</td>
</tr>
</tbody>
</table>
Table 4-2  (Cont.) HTML Files Used by Self-Care Manager

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>header.html</td>
<td>HTML code that you can use as a header on every JSP page by including the file name in the JSP.</td>
</tr>
<tr>
<td>index.html</td>
<td>The default account creation home page. Customers see this page first when they go to your Web site.</td>
</tr>
</tbody>
</table>

Other Files Used by Self-Care Manager

Self-Care Manager also uses these files:

- **Style sheet**: The HTML version of the JSPs use the style sheet `infranet_general.css`.
- **GIF files**: The HTML version of the JSPs use several GIF files. Most are stored in `htmlui_en/graphics`, others are stored in `htmlui_en`. 
Setting up Welcome Messages to Customers

This document describes how to send introductory messages to customers by email or on a Web page, in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Sending a Welcome Email
- Setting Up a Welcome Web Page
- Using Variables to Include Customer Data in Welcome Emails and Web Pages
- Using Multiple Welcome Emails and Web Pages

See also:

- List of Customer Management Features
- BRM Concepts

Sending a Welcome Email

The default email message is located in `BRM_home/sys/cm/welcome/default.welcome`. Use any text editor to modify the text. Before you can send the message you must:

- Configure the Email Data Manager (DM) that sends the message. See “Sending Email to Customers Automatically”.
- Enable the email on the BRM server.

To enable the welcome message:

1. Open the Connection Manager (CM) configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the value of the `new_account_welcome_msg` entry to 1.
3. Save and close the file.

The new value becomes effective immediately and applies to the next account created. The CM does not need to be restarted.

See also:

- Changing the Welcome Email Subject Line
- Changing the Welcome Email Sender Address
- Specifying the Welcome Email Location
- Disabling the Welcome Email
Changing the Welcome Email Subject Line

The subject line for the welcome email message is by default “Welcome to the Internet.” To change this, edit the PCM_OP_CUST_POL_POST_COMMIT policy opcode.

Changing the Welcome Email Sender Address

The sender address is the email address that your customer sees as the sender address.

To change the welcome message sender address:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. (Optional) Change the sender entry. This entry changes the part of the email address that precedes the at sign (@); for example:
   
   `sender@your_domain.com`
   
   The default is postmaster.
3. Change the domain entry. This entry changes the part of the email address that follows the at sign (@), for example:
4. Save and close the file.
   
   `sender@your_domain.com`

You do not need to restart the CM to enable this entry.

Specifying the Welcome Email Location

If you move the welcome message file from the default location, you must specify the new location.

To specify a new welcome message location:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the value of the welcome_dir entry.
3. Save and close the file.

The new value becomes effective immediately and applies to the next account created. The CM does not need to be restarted.

Disabling the Welcome Email

If the welcome message is enabled, but the Email DM is not running, you will see an error in the CM log file. If you do not run the Email Data Manager, disable the welcome message.

To disable the welcome message, do one of the following:

- Change the value of the `new_account_welcome_msg` entry in the CM configuration file (`BRM_home/sys/cm/pin.conf`) to 0 (default).
- Rename the default message file.
• Remove write permission on the default message file.

Setting Up a Welcome Web Page

The introductory Web page is an HTML file that you can display during Web-based account creation. The introductory message is not implemented by default. To display an introductory message, you must customize your automatic account creation method to call the PCM_OP_CUST_POL_GET_INTRO_MSG policy opcode.

Use any text editor to edit the introductory message. The default introductory message is located in `BRM_home/sys/cm/intro/default.intro`.

If you move the introductory message file from the default location, you must specify the new location.

To specify a new introductory message location:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the value of the `intro_dir` entry.
3. Save and close the file.

The new value becomes effective immediately and applies to the next account created. You do not need to restart the CM to enable this entry.

Using Variables to Include Customer Data in Welcome Emails and Web Pages

You can use variables to insert information based on the customer’s input into an email or an HTML page. For example, to display the package that the customer selected, you use the `${price_plan}` variable in the introductory message:

```
You have selected the ${price_plan} package
```

If the customer selects the Basic package, the message says “You have selected the Basic package.”

Table 5-1 shows the default variables:

<table>
<thead>
<tr>
<th>Use this variable</th>
<th>To insert this</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>${price_plan}</code></td>
<td>Package name</td>
<td>Basic</td>
</tr>
<tr>
<td><code>${plan_description}</code></td>
<td>Package description</td>
<td>Monthly broadband access</td>
</tr>
<tr>
<td><code>${promo_code}</code></td>
<td>Promotional code</td>
<td>Broadband1</td>
</tr>
</tbody>
</table>

You can define more variables by customizing the PCM_OP_CUST_POL_GET_INTRO_MSG policy opcode.

Using Multiple Welcome Emails and Web Pages

You can create multiple welcome messages to display to specific groups of customers; for example, you might want different messages for each language.
Which message is displayed is based on a value entered during account creation.

**Note:**

This value is stored in the PIN_FLD_AAC_SOURCE field in the account object.

For example, you could identify which message to send by using the IP address that the user logged in to. In that case, an introductory message file name might be:

**156.151.11.1700.intro**

In this example:

- **156.151.11** is the IP address
- **1700** is the port number
- **intro** is the suffix
Sending Email to Customers Automatically

This document describes how to set up, configure, and start the Oracle Communications Billing and Revenue Management (BRM) Email Data Manager (DM), \texttt{dm\_email}. The Email DM is a process that enables you to send customer notifications and invoices.

After the Email DM is running, you need to set up event notification to determine the events or times that trigger a customer email.

See also:
- List of Customer Management Features
- BRM Concepts

Configuring Email Data Manager

To configure Email Data Manager:

1. Open the CM configuration file (\texttt{BRM\_home/sys/dm\_email/pin.conf}).
2. Make sure these entries exist in the file:
   - \texttt{dm\_pointer}
   - \texttt{fm\_module}
   - \texttt{em\_db}
3. If you changed the file, save it, and stop and restart the CM.

Starting the Email Data Manager

To start the Email DM, run this command:

\texttt{pin\_ctl start dm\_email}

Configuring sendmail Options

The Email DM (\texttt{dm\_email}) uses the \texttt{sendmail} program.

By default, the DM uses the following \texttt{sendmail} command:

\texttt{/usr/lib/sendmail -t}

This default should be sufficient for most users. But you can configure BRM to use different command-line options for \texttt{sendmail}. 
Note:
Do not change the sendmail configuration unless you are an experienced sendmail user.

1. Open the Email DM configuration file (BRM_home/sys/dm_email/pin.conf).
   BRM_home is the directory in which the BRM server software is installed.
2. Edit the unix_sendmail_command entry. You can specify different options for sendmail, but you cannot substitute a different program for sendmail.
3. Save the file.
4. Stop and restart the Email DM.
Managing Passwords and Login Names

This document describes how to manage Oracle Communications Billing and Revenue Management (BRM) customer authentication, including login names, passwords, and security codes.

Topics in this document:

• About Login Names and Passwords
• Using Security Questions
• Assigning Passwords Automatically
• Defining Email Login Names
• Detecting Duplicate Logins

See also:

• List of Customer Management Features
• BRM Concepts

About Login Names and Passwords

By default, all services require a login name and password. For services such as telephony where a customer doesn't use a login and password, logins and passwords can be generated automatically for internal use.

• By default, the login name can be a minimum of 1 character and a maximum of 255 characters.
• By default, the password must be a minimum of 1 character and a maximum of 255 characters.
• Login names are unique and can be assigned to only one account.

You can change the minimum and maximum login name and password lengths by using the Field Validation Editor. See "Specifying How to Validate Customer Contact Information".

All login names and passwords are associated with a service; for example, the broadband service (/service/ip). Customer service representatives (CSRs) use a login name and password to log in to the admin_client service.

BRM uses two types of customer passwords:

• Customers use service passwords, such as the password that a customer provides when logging in to an broadband connection, to access a broadband service.
• Customers use account passwords for non-IP access, such as accessing a Web page.

By default, account passwords are stored in the database in an encrypted format; service passwords are not.
To customize password encryption, use the
PCM_OP_CUST_POL_ENCRYPT_PASSWD,
PCM_OP_CUST_POLCOMPARE_PASSWD, and
PCM_OP_CUST_POL_DECRYPT_PASSWD opcodes. See BRM Opcode Guide.

For more information about encryption, see "About Encrypting Information" in BRM
Developer's Guide.

About Telco Service Logins and Passwords

When you create an account that uses telco services, the customer ID and password
are generated automatically. Therefore, a CSR does not need to enter an ID and
password at account creation or when adding a telco service.

Note:

Internally, the customer ID is the same as the login name.

• To ensure that a unique ID is generated, the default ID is a unique string
  composed of the following elements:
  – A timestamp generated by the Connection Manager (CM) that was used for
    creating the account.
  – The process ID (PID) of the CM.
  – The thread ID of the CM (always 1).
  – The CM host name.

For example:
269-20011128-095216-7-22493-1-host_name

When an ID is needed: for example, for Web-based account management: the
customer enters their MSISDN or IMSI. Applications can retrieve the MSISDN or
IMSI from the customer's service objects. (Customers can also enter the ID.)

Note:

When using an MSISDN or IMSI as a login, the customer must enter the
full number with no punctuation, such as 014085551212.

To customize how IDs are generated, you customize the
PCM_OP_CUST_POL_PREP_LOGIN policy opcode.

• The default password is password. You cannot change the password when a
  service is being added, but you can change it later.

To customize how passwords are generated, you customize the
PCM_OP_CUST_POL_PREP_PASSWD policy opcode.
Using Security Questions

You can specify two security questions for a customer. When a customer calls a CSR, the CSR asks the customer the security question, which is displayed in Customer Center or Billing Care.

Unlike service passwords, security codes are not validated by BRM; therefore, you cannot enforce properties such as the number of characters in a security code.

Assigning Passwords Automatically

You can set up account creation to do either of the following:

- Require the customer to specify a password. This is the default.
- Generate a password automatically for the customer.

To generate a password for the customer, you must supply the algorithm for generating passwords. To do so, customize the PCM_OP_CUST_POL_PREP_PASSWD policy opcode.

Defining Email Login Names

Note:

Changing the customer's email login name also changes the customer's email address. Before changing a login name, make sure the customer wants to change the email address.

To change the requirements for case-sensitivity in email login names, customize the PCM_OP_CUST_POL_PREP_LOGIN and PCM_OP_CUST_POL_VALID_LOGIN opcodes. See BRM Opcode Guide.

The default email login requirements are:

- The email login must use all lowercase characters.
- The email login must include the domain, in this format: login@domain

For example: francisco@oracle.com

Detecting Duplicate Logins

By default, BRM does not check for duplicate logins. This means that more than one customer can log in to a service by using the same name. To check for duplicate logins, customize the PCM_OP_ACT_POL_SPEC_VERIFY policy opcode.
This document describes how to record login failures in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

- Recording Login Failures
- Specifying the Account That Records Login Failures
- Viewing Login Failures

See also:

- List of Customer Management Features
- BRM Concepts

**Recording Login Failures**

You can record login failures to detect possible service problems or fraud. For example, duplicate logins might indicate fraudulent usage. You can specify which login failures to record.

You can record login failures for any reason except using an unknown login name. For example, you can record events when customers try to:

- Log in with the wrong password
- Log in to an inactive service
- Log in after exceeding their credit limit

You can also record successful logins, although doing this slows BRM performance.

To record login failures, specify the types of login failures you want to record in the `pin_verify` file and then load those preferences into the BRM database with the `load_pin_verify` command.

---

**Note:**

The `load_pin_verify` utility requires a configuration file. See “Creating Configuration Files for BRM Utilities" in *BRM System Administrator’s Guide*.

---

To specify login failure types:

1. Edit the `BRM_home/sys/data/config/pin_verify` file, where `BRM_home` is the directory in which the BRM server software is installed. The `pin_verify` file includes examples and instructions.
   
   You can make these types of changes:
• Edit a list of predefined types of login failures to specify which events you want to record and to modify their descriptions.
• Add custom types of login failures you want to record.

**Note:**

When you run load_pin_verify, it overwrites the existing preferences for recording customer login failures. If you are updating your preferences, you cannot load new preferences only. You must load complete sets of preferences each time you run the load_pin_verify utility.

2. Save the file.
3. Use the following command to run load_pin_verify utility:

   ```
   load_pin_verify pin_verify_file
   ```

   For more information, see "load_pin_verify".

4. Stop and restart the CM.

To verify that the network elements were loaded, you can display the /config/verify object by using Object Browser, or by using the robj command with the testnap utility. (See "Reading an Object and Writing Its Contents to a File" in BRM Developer’s Guide.)

---

**Specifying the Account That Records Login Failures**

BRM logs verification events against the root account by default. If you want to use a different account, do the following:

1. (Optional) Create an account specifically for logging verification events. You can create a CSR account or a dummy account.
   
   If you create a dummy account, you should:
   
   • Use a dummy package that includes no charge pricing and applies to accounts but not to any service.
   • Use the internal payment method.
   • Use any name and address you want. You have to enter something for name and address to create an account.

2. Open the CM configuration file in BRM_home/sys/cm.

3. Add this entry to the end of the file:

   ```
   fm_act account_name database_number /account 1
   ```

   where

   • `account_name` is the name of the account to which verification logging should go.
   • `database_number` is the database number of the BRM database. By default, this number is 0.0.0.1.
This example sets an account called **verify_acct** to log verification events:

- `fm_act verify_acct 0.0.0.1 /account 1`

4. Save the file.
5. Stop and restart the CM.

**Viewing Login Failures**

To view login failures, use Billing Care or Customer Center to open the root account or the account you specified in the CM configuration file.
Managing Account and Service Status

This document describes how to change account and service status in Oracle Communications Billing and Revenue Management (BRM).

See BRM Concepts for basic information about account status.

Topics in this document:

• Automatically Inactivating and Closing Accounts
• Scheduling Status Changes in Advance
• Changing Start Times for Reactivated Charge and Discount Offers
• Allowing Active Services with Inactive Accounts
• Inactivating and Closing Accounts in Hierarchies
• Reusing Login Names and Passwords from Closed Accounts or Canceled Services

See also:

• List of Customer Management Features
• Backdating Subscription Actions

Automatically Inactivating and Closing Accounts

By default, BRM automatically inactivates accounts in the following situations:

• A credit card payment fails because a nonvalid credit card is used.
• A credit card validation fails, and the item is 30 or more days past due.
• The account is a child account with a nonpaying bill unit (/billinfo object), and the parent account is inactivated.

Note:

By default, account status is not changed when a credit limit is reached. However, also by default, service authorization requires that the credit limit has not been reached, so reaching a credit limit prevents a customer from using a service.

To change these defaults, you must customize the following policy opcodes:

• PCM_OP_PYMT_POL_COLLECT
• PCM_OP_ACT_POL_EVENT_LIMIT
• PCM_OP_ACT_POL_SPEC_VERIFY
Scheduling Status Changes in Advance

In Billing Care or Customer Center, you can schedule account and service status changes for a future date. You use a daily billing utility, `pin_deferred_act`, to change the status automatically on the scheduled date.

You can schedule the following types of actions to occur on a future date by:

- Changing the status of an account or service.
- Adding an account to a hierarchy.
- Removing an account from a hierarchy.
- Moving an account between hierarchies.

After an action is deferred, you can display, execute, reschedule, and cancel the deferred action.

Scheduling account reactivation works differently for inactivating and closing:

- When you inactivate an account or service, you can set a future date to reactivate the account or service, or you can reactivate the account or service manually.
- When you close an account or service, you cannot schedule reactivation. Closed accounts and services must be reactivated manually.

Use the `pin_bill_day` script to run the `pin_deferred_act` utility daily. To ensure that the account status is correct before running billing, it is the first billing utility run by the `pin_bill_day` script. See "pin_deferred_act".

Changing Start Times for Reactivated Charge and Discount Offers

By default, when inactive charge and discount offers are reactivated, their purchase, usage, and cycle start times are changed to the current (reactivation) date. You can use the Connection Manager (CM) configuration file `change_start_time_on_activation` entry to change this behavior so that when inactive charge and discount offers are reactivated, the original purchase, usage, and cycle start times are kept.

This setting does not affect charge offers whose purchase is deferred to a later date. If the charge offer is reactivated before the purchase date, the original purchase, usage, and cycle start times are kept, even if this option is enabled.
Note:

The purchase and cycle start times determine the date on which the purchase and cycle charges begin to be applied to the charge and discount offers.

To keep the original purchase, usage, and cycle start times:

1. Open the CM configuration file (BRM_home/sys/cm/pin.conf). BRM_home is the directory in which the BRM server software is installed.

2. Add the following entry:

   -fm_bill change_start_time_on_activation 0
   • 0 keeps the original purchase, usage, and cycle start times.
   • 1 changes the purchase, usage, and cycle start times to the reactivation date. This is the default.

3. Save and close the file.

4. Stop and restart the CM.

Allowing Active Services with Inactive Accounts

By default, inactivating an account inactivates all services owned by the account. You can use the allow_active_service_with_inactive_account CM pin.conf file entry to allow inactive accounts to have active services.

To allow active services with inactive accounts:

1. Open the CM configuration file (BRM_home/sys/cm/pin.conf).

2. Add the following entry.

   -fm_cust allow_active_service_with_inactive_account 1
   • 0 prohibits active services with inactive accounts. This is the default.
   • 1 allows inactive accounts to have active services.

3. Save and close the file.

4. Stop and restart the CM.

Inactivating and Closing Accounts in Hierarchies

Changing the status of an account in an account hierarchy changes the status of all accounts inside and outside the account hierarchy that have at least one nonpaying bill unit whose paying parent bill unit is owned by the initially changed account.

For more information, see “How Account Status Changes Affect Hierarchies” in BRM Managing Accounts Receivable.

You cannot prevent the status of a nonpaying child bill unit from changing when its parent bill unit's status changes.
Reusing Login Names and Passwords from Closed Accounts or Canceled Services

By default, you can not reuse the login name of a closed account or canceled service, although BRM has no restriction on reusing passwords.

To reuse login names, set up BRM to automatically modify the login name when you close an account. For example, you could have the string "CLOSED#" prepended to the login name.

You can customize the PCM_OP_CUST_POL_PREP_STATUS policy opcode to automatically add characters to a login name when an account's status changes.
Managing Credit Cards During Account Creation

This document describes how to manage credit cards during account creation in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:

• Charging Customers at Account Creation
• Validating Credit Cards at Account Creation
• Avoiding Credit Card Revalidation
• Allowing Account Creation without a Credit Card
• Increasing Account Creation Speed When Paymentech Is Offline
• Specifying the Account That Records Credit Card Validations

See also:

• List of Customer Management Features
• BRM Configuring and Collecting Payments
• BRM Concepts

Charging Customers at Account Creation

By default, BRM charges for purchase fees and the first cycle forward fees at account creation. When a customer creates an account, BRM uses the credit card processor to authorize the payment for the fees. The next time you run billing, the pin_deposit_cc billing utility deposits the credit card payment. (You should run the pin_deposit_cc utility as part of the pin_bill_day script. See "About the Billing Utilities" in BRM Configuring and Running Billing.)

The total due amount for the account is charged immediately and the payment is allocated immediately to all open bill items. Therefore, after the account is created, it will have no pending amount due and no unapplied payments.

To determine how to charge customers at account creation:

1. Open the CM configuration file (BRM_home/sys/cm/pin.conf).
2. Change the value of the -fm_pymt_pol cc_collect entry:
   • To enable credit card collection, enter 1.
   • To disable credit card collection, enter 0.
3. Save the file.

You do not need to restart the CM to enable this entry.

Use the PCM_OP_PYMT_POL_SPEC_COLLECT policy opcode to customize whether to charge the customer immediately for all or part of the current account...
balances during account creation or to defer the charges to a later date. You can also specify whether to validate the charges. See BRM Opcode Guide.

Which Fees Are Charged at Account Creation

Only fees for charge offers that are purchased as part of the account creation process are charged and billed immediately. All other charge offer fees, even for those charge offers purchased the same day as account creation, are charged in the first billing cycle. For example, if a customer creates an account and then purchases an additional charge offer an hour later, the additional purchase fee is charged at the end of the current billing cycle.

How General Ledger Revenue Is Reported for Account Creation Charges

Even though charges collected at account creation have been paid, the BRM general ledger (G/L) interface categorizes those charges as unbilled revenue until the first billing cycle.

Validating Credit Cards at Account Creation

By default, BRM validates credit cards by checking the customer's name and address and the credit card number. You might want to turn validation off if the connection to the credit card processor is offline.

A customer can create an account even when:

- The credit card transaction processing service is unavailable.
- The ZIP code is valid but the street address is wrong.

By default, a customer cannot create an account if:

- The wrong ZIP code was entered.
- The credit card is not valid.
- The address was not entered or could not be read by the credit card processor.

Note:

If an account is created without validating the credit card, the account will not be charged for any fees when it is created.

To enable or disable credit card validation, do the following:

1. Open the CM configuration file (BRM_home/sys/cm/pin.conf).
2. Change the value of the cc_validate entry:
   - To enable credit card validation, enter 1.
   - To disable credit card validation, enter 0.
3. Save the file.
Avoiding Credit Card Revalidation

You can specify the credit card revalidation interval. For example, you can allow an interval of one hour before a credit card needs to be revalidated. This reduces the cost of credit card validations and allows customers who use the same credit card to create multiple accounts without having to validate the credit card more than once.

To specify the credit card revalidation interval:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the value of the `cc_revalidation_interval` entry. The default is 3600 seconds (one hour).
3. Save the file.

You do not need to restart the CM to enable this entry.

Allowing Account Creation without a Credit Card

You can allow your customers to create accounts and sign up for services without providing a credit card number when they create an account. This means that you can offer your customers a number of free days of service before they have to provide a credit card number.

To implement this feature, create the accounts by using the Undefined payment method. You can change the payment method to credit card or invoice after the free period has ended. You can also use the PCM_OP_CUST_SET_BILLINFO opcode to set the payment method. See `BRM Opcode Guide`.

Increasing Account Creation Speed When Paymentech Is Offline

If you know that your connection to Paymentech will be offline, you can speed up account creation by not allowing timeouts. Instead, a connection to Paymentech results in a “no answer” error immediately. By default, account creation can occur even though there is a “no answer” error. Also, a “no answer” error does not create checkpoint records, so you do not have to resolve the transaction.

Note:

If you use this option, you cannot process credit card transactions by using the `pin_collect` or `pin_deposit` utilities. You must wait and run billing when the Paymentech connection is restored.
Note:

If the credit card payment service is not available and you still want to create accounts, you must isolate those accounts for later credit card authorization. Modify the PCM_OP_PYMT_POL_VALIDATE policy source file either to save a list of permissive account creation or to send email to the system administrator. Alternatively, you can write a simple application to periodically check accounts and flag the accounts that have been created without verification.

1. Open the Paymentech DM configuration file (*BRM_home*/sys/dm_fusa/pin.conf).
2. Edit the **online_proto** entry:
   - Enter `linkdown` to disable timeouts and report "no answer" for all connections.
   - Enter `socket` to enable the connection to function normally.
3. Edit the **batch_proto** entry:
   - Enter `linkdown` to disable timeouts and report "no answer" for all connections.
   - Enter `socket` to enable the connection to function normally.
4. Save the file.
5. Stop and restart the Paymentech DM.

### Specifying the Account That Records Credit Card Validations

To validate credit cards, BRM needs an account. By default, BRM logs credit card validations against the root account during account creation.

If you want to use a different account, do the following:

1. Open the Connection Manager (CM) configuration file (*BRM_home*/sys/cm/pin.conf). *BRM_home* is the directory in which the BRM server software is installed.
2. Change the account number in the following entry:
   ```
   - fm_pymt_pol validate_acct database_number /account 1
   ```
   where `database_number` is the database number of the BRM database; by default, this is 0.0.0.1.
3. Save the file.

You do not need to restart the CM to enable this entry.

To customize this feature, use the PCM_OP_PYMT_POL_SPEC_VALIDATE policy opcode. See *BRM Opcode Guide*. 

---

*ORACLE*
Changing a Customer's Billing Configuration

This document describes how to manage changes to customer billing information, such as changes to billing dates, credit limits, and payment methods.

Topics in this document:

• Changing a Customer's Billing Day of Month
• Changing a Customer's Accounting Type
• Changing an Account's Bill Unit to Nonpaying
• Changing a Closed Child Account's Bill Unit to Nonpaying

See also:

• List of Customer Management Features
• BRM Concepts

Changing a Customer's Billing Day of Month

Use Billing Care or Customer Center to change the billing day of month.

A change to a billing day of month (DOM) takes effect in the next billing cycle. This means that there will be a partial billing cycle to handle the difference in days between the end of the current billing cycle and the start of the new billing cycle. For more information, see “Specifying How to Handle Partial Accounting Cycles” in BRM Configuring and Running Billing.

When you change the billing date and create a partial billing cycle, BRM prorates all cycle fees for that cycle.

Changing a Customer's Accounting Type

A bill unit's accounting type can be changed at any time. However, BRM does not validate the change nor take any actions other than changing the accounting type in the next billing cycle. You must ensure that the impact of any accounting type changes do not confuse your customers. For example:

• If the accounting type is changed from open item accounting to balance forward accounting, the customer’s next bill would include all open and unpaid items. Your customer should be informed that the bill now includes any past due charges from previous billing cycles.

• If the accounting type is changed from balance forward accounting to open item accounting, the customer’s next bill would not include any unpaid items. Your customer should be informed that charges from previous bills are still past due, even though they do not appear on the current bill.
Changing an Account's Bill Unit to Nonpaying

If you change a child account bill unit's payment method to nonpaying after creating the account, the billing information for the nonpaying bill unit changes to match the parent account. A nonpaying bill unit must use the same currency, billing date, billing frequency, and accounting type as its parent account. If the accounts use two currencies, the parent account and the nonpaying bill unit in the child account must use the same primary currency.

Changing a Closed Child Account's Bill Unit to Nonpaying

You can change a closed child account bill unit's payment method to nonpaying (subordinate) if either of the following conditions is true:

- The closed child account was billed for all the previous cycles.
- The closed child account was not billed because you have disabled billing of closed accounts and the child account's total balance due for the bill unit is zero. See "Suspending Billing of Closed Accounts" in BRM Configuring and Running Billing for more information about disabling the billing of closed accounts.
Managing Purchased Charge Offers

This document describes how to manage Oracle Communications Billing and Revenue Management (BRM) customers’ purchased charge offers.

Topics in this document:

- Canceling Charge Offers
- Enabling Charge Offer Purchases from Closed or Inactive Accounts
- Changing Charge Offer Quantity

See also:

- List of Customer Management Features
- BRM Concepts

Canceling Charge Offers

You can cancel a charge offer immediately or backdate the cancellation to a date earlier than the current date. You can also cancel a charge offer in the future by modifying the charge offer end times.

See these topics for more options:

- Canceling Charge Offers Without Charging a Cancel Fee
- Including Event Adjustments in Charge Offer Cancellation Refunds
- Calculating Conditional Discount Offers When Canceling a Charge Offer
- Deleting Canceled Charge Offers

You can cancel individual charge offers or cancel all charge offers in a bundle at one time by canceling the bundle.

Note:

Deferred cancellation is available only for the entire quantity of the charge offer in the account.

If a charge offer's cycle forward fee allows proration, the customer is credited for any cycle forward fees that have been charged for but not used. BRM prorates based on the pricing in effect at the time the charge offer was purchased.

When you close an account, all of the charge offers it owns are canceled.
Canceling Charge Offers Without Charging a Cancel Fee

You can specify the amount of time after a purchase that a charge offer can be canceled without charging the customer. For example, you might want to specify a charge offer cancel tolerance of 30 minutes in case a CSR assigns the wrong charge offer to an account and needs to cancel it without charging the customer.

To specify a cancellation tolerance:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the value of the `cancel_tolerance` entry.
   - `0` (Default) to always charge.
   - Any other value entered in `cancel_tolerance` is in minutes.
   For example (15 minutes):
     - `fm_bill cancel_tolerance 15`
3. Save and close the file.
4. Stop and restart the CM.

Including Event Adjustments in Charge Offer Cancellation Refunds

By default, when a charge offer is canceled, BRM does not apply event adjustments when calculating the refund amount.

To configure BRM to include event adjustments in charge offer cancellation refunds, run the `pin_bus_params` utility to change the `EventAdjustmentsDuringCancellation` business parameter. For information about this utility, see `BRM Developer's Guide`.

To include event adjustments in charge offer cancellation refunds:

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 `0` to `1`:
   ```xml
   <EventAdjustmentsDuringCancellation>0</EventAdjustmentsDuringCancellation>
   ```
4. Save the file as `bus_params_subscription.xml`.
5. Load the XML file into the BRM database:
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*.

## Calculating Conditional Discount Offers When Canceling a Charge Offer

When canceling a charge offer, discounts that are based on conditions cannot be reliably calculated. When a charge offer is canceled, the discount calculated during monthly charges is recalculated because the condition values applied during the monthly charge may be different from the values applied at the time of charge offer cancellation.

For example, suppose a customer receives a 5% discount when the total monthly charge is less than $100, but receives a 10% discount when the total monthly charge is greater than $100. At charge offer purchase time, the total monthly charge is just the purchase amount, which is likely to be less than $100, so a 5% discount is applied. If the charge offer is canceled in the same month or in a subsequent month, but the total monthly charge is greater than $100, a 10% discount is applied as a prorated charge (if the discount is set to be prorated). This occurs because the charge offer cancellation occurs before the end of the month.

### Note:

Conditional discounts are rarely used for purchase and cancellation events.

To ensure that conditional discounts are reliably calculated when canceling a charge offer, do one of the following:

- When you define a discount, select options to prorate the balance impact for the amount to ensure there is no refund at the time of charge offer cancellation.
- If the discount calculated at charge offer cancellation is not correct and the customer calls to get a resolution, the CSR can manually calculate the correct discount and the prorated refund amount.

## Deleting Canceled Charge Offers

You might want to delete canceled charge offers from your database (for example, to improve performance by reducing the amount of data stored in the database).

Oracle recommends that you not delete canceled charge offers and discount offers because other external systems might also use them.
Note:

- Do not delete canceled charge offers if you use delayed billing. By default, BRM does not delete `/purchased_product` objects (charge offer instances) when you cancel them. When BRM is set up to use delayed billing, BRM requires information about the canceled charge offers to charge for the delayed events that were generated before the charge offer was canceled. To charge for events for canceled charge offers, the canceled charge offer instances must persist as `/purchased_product` objects, which is the default behavior.

- Do not delete canceled charge offers if you rerate events. Events that use deleted charge offers cannot be rerated.

- You cannot delete a charge offer if provisioning tags are defined for that charge offer. Charge offers with provisioning tags are updated by the provisioning system and therefore must remain in the BRM database.

To automatically delete `/purchased_product` objects when a charge offer is canceled:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the value of the `keep_cancelled_products_or_discounts` entry.
   - `0` deletes the canceled `/purchased_product` objects.
   - `1` (default) keeps the deleted charge offers in the `/purchased_product` objects.
3. Save and close the file.
4. Restart the CM.

Enabling Charge Offer Purchases from Closed or Inactive Accounts

By default, closed and inactive accounts cannot purchase charge offers. You may, however, want to make such purchases possible.

For example, you may want CSRs to create accounts that are closed, then purchase charge offers for the account. This allows the customer to verify that the charge offer is correct before charging the account. When the account is made active, the customer is charged.

When you purchase a charge offer for inactive or closed accounts, the offer's status must be inactive. If the offer's status is active, rating modules rate and apply charges for the offer. If you do not want charges to be applied, change the offer's status to inactive at the time of purchase, or use PDC or Pricing Center to set the offer's status to inactive in the bundle. When the account is activated, change the offer's status to active for charges to be applied.

To enable charge offer purchases from closed or inactive accounts:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
In the XML file, change the value for \texttt{DealPurchaseForClosedAccount} to \texttt{enabled}:
\[
\textless \text{DealPurchaseForClosedAccount}\texttt{enabled}\textgreater /\text{DealPurchaseForClosedAccount}
\]

Load the XML file into the BRM database:
\[
\texttt{pin\_bus\_params\ bus\_params\_billing.xml}
\]

Stop and restart the CM.

(Multischema systems only) Run the \texttt{pin\_multidb} script with the \texttt{-R CONFIG} parameter. For more information, see \textit{BRM System Administrator's Guide}.

Changing Charge Offer Quantity

You can change the quantity of a purchased charge offer; for example, a charge offer that gives one hundred included minutes as a sign-up bonus.

If you increase the quantity of a charge offer that has a discount offer on purchase fees, that discount offer is applied to the entire purchase rather than to each unit of the charge offer that is added to the account. For example, if you increase the quantity of a charge offer in an account from one to five and the discount offer on purchase fees is $5, the customer receives a discount offer of $5, not $20.
Managing Purchased Discount Offers

This document describes how to manage Oracle Communications Billing and Revenue Management (BRM) customers' purchased charge offers.

Topics in this document:

- Setting Discount Offer Status
- Setting Discount Offer Purchase, Cycle, and Usage Start and End Times
- Enabling Mutually Exclusive Discount Offers
- Configuring Remaining Charge Cycle Discounting
- Canceling Discount Offers

See also:

- List of Customer Management Features
- BRM Concepts

### Setting Discount Offer Status

The status of a discount offer can be **Active**, **Inactive** or **Canceled**. A discount offer is not valid when it is inactive. The discount offer does not apply to any user events generated while it is inactive.

When you change the status of a discount offer, you specify the new status and the reason for the status change.

Discount offer status can change in the following cases:

- When you purchase a discount offer: you can set the status to **Active** or **Inactive** in the case of deferred purchase.
- When you purchase an inactive discount offer: you can later reactivate it by setting its status to **Active**.
- When you cancel a discount offer: you set the discount offer status to **Canceled**. See “Canceling Discount Offers”.
- When you change the status of an account or service that owns a discount offer: you change the status of the discount offer to the same status; when you close an account, you cancel any discount offers it owns.

### Setting Discount Offer Purchase, Cycle, and Usage Start and End Times

You specify a discount offer's validity period for the account that purchases the discount offer by setting the purchase start and end times. The cycle and usage start
and end times specify when to start and stop charging cycle forward fees and rating usage events by using the discounted charge.

You can modify a discount offer's purchase, usage, and cycle start and end times at the following times:

- When purchasing a discount offer.
- When canceling a discount offer. By default, the discount offer's purchase, cycle, and usage end times are set to the cancellation time. You can change the date the discount offer expires by modifying the discount offer's end time.
- When changing the status of a discount offer.

**Note:**
Do not change a discount offer's purchase, cycle, or usage start time after the discount has been applied to the account; otherwise, the discount will be applied incorrectly.

BRM does not allow you to backdate the discount offer's purchase, usage, or cycle end date prior to the discount offer's purchase start date or prior to the G/L posting date.

The cycle and usage periods must start after the purchase period start time and end before the purchase period end time.

If you configure BRM for time stamp rounding, BRM rounds all start and end times to 00:00:00 hours.

## Enabling 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 configure discount offer exclusions in either PDC or Pricing Center, and in a BRM server configuration file.

By default, discount exclusions are disabled in BRM. You can enable them by running the `pin_bus_params` utility to change the `ValidateDiscountDependency` business parameter. For information about this utility, see *BRM Developer's Guide*.

When you enable discount exclusions, you have the following options:

- You can enable both discount offer-to-discount offer exclusion and discount offer-to-package exclusion only simultaneously.
- With discount offer-to-package exclusions, you can disable the exclusion when a discount offer is purchased, but allow it to be enforced at run time when a discount is applied to a charge.

To enable discount exclusions in BRM:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   
   ```bash
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
3. Find the following line:
4. Change `disabled` to any of the following:
   - `discToDiscExcl`: Enable discount offer-to-discount offer exclusions.
   - `discToPlanExcl`: Enable discount offer-to-package exclusions.
   - `enableBothExcl`: Enable both discount offer-to-discount offer and discount offer-to-package exclusions.
   - `disableDiscToPlanExclAndNoPurTimeValidation`: Disable exclusions between packages and discount offers systemwide. When this flag is set, at purchase time no dependency validations are performed.
   - `enableBothExclAndNoPurTimeValidation`: Enable both discount offer-to-discount offer and discount offer-to-package exclusions, and do not support dependency validations at purchase time.
   - `returnOnFirstExcl`: Use the first mutually exclusive discount offers or discount offer/package if BRM finds any such conflicts.

   **Note:**

   BRM uses the XML in this file to overwrite the existing `billing` instance of the `/config/business_params` object. If you delete or modify any other parameters in the file, the changes affect the associated aspects of the BRM billing configuration.

5. Save the file as `bus_params_billing.xml`.

6. Load the XML file into the BRM database:
   ```
   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.

---

**Configuring Remaining Charge Cycle Discounting**

Remaining charge cycle discounting is used in the following situation:

1. A discount is applied to a recurring fee.

2. A second discount is applied mid-cycle. In this case, the second discount might be applied to the original charge instead of to the remaining charge.

3. If remaining charge cycle discounting is enabled, the original discount is backed out and the two discounts are evaluated based on priority.

If a second discount is purchased mid-cycle, the original discount is backed out and the two discounts are evaluated based on priority.

This process has the following consequences and limitations:

- Although this process is run for all cycle fee discounts that are applied mid-cycle, it can change the outcome only for cycle fee discounts that are set to Remaining Charge, and are prorated.
• This process locks the account involved in the discount offer applied during the entire process or refunding and reapplying discounts. The lock is released at the end of the transaction.

• If the discount is shared, all members of the discount sharing group are locked, but only if the sharing start time is set to start when the discount sharing group is created or a member is added (the propagate_discount entry is set to 1 in the Connection Manager (CM) pin.conf file). Locks are released for all members at the end of the transaction.

• This process does not retroactively change balances. When a discount that grants a noncurrency balance is applied mid-cycle, calls rated based on the noncurrency balance can be incorrect.

For example, a cycle fee discount grants 100 minutes and the subscriber uses all 100 minutes in the first week. The discount is then canceled mid-month. The discount amount is reevaluated, resulting in an adjusted grant of 50 minutes. However, the minutes already used by the subscriber beyond the new grant amount are not recovered because the calls have already been rated. In this case, you can rerate the calls for the account.

• This process uses the discount exclusion rules that are in effect at the time of the evaluation. For example, if an exclusion rule changes on the 15th of the month and discounts are reevaluated on the 20th, the exclusion rule that took effect on the 15th is used when reevaluating discounts for the entire cycle.

By default, remaining charge discounting of cycle fees is disabled. You can enable this process by running the pin_bus_params utility to change the SequentialCycleDiscounting business parameter. For information about this utility, see BRM Developer's Guide.

To enable remaining charge discounting of cycle fees:

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:

   <SequentialCycleDiscounting>enabled</SequentialCycleDiscounting>

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.

Canceling Discount Offers

You typically cancel a discount offer when you cancel the bundle in which it is bundled or when you close the account or service that owns the discount offer. You can also cancel a discount offer immediately or backdate the cancellation to a date earlier than the current date. A discount offer gets canceled automatically when the discount offer's purchase end date has been reached.
When you cancel a discount offer, you also prorate and charge cycle forward fees that may have been discounted for the period during which the discount offer is no longer valid.

To cancel a discount offer, you must specify the quantity to cancel:

- If the quantity is the same as the quantity purchased by the account or service, BRM cancels the discount offer.
- If the quantity is less than the quantity purchased by the account or service, BRM does not cancel the discount offer; it subtracts that quantity from the internal count variable within the `/purchased_discount` object that represents the discount offer.

By default, when you cancel a discount offer, BRM retains the `/purchased_discount` object that describes the discount offer with a status of `Canceled`. You can specify whether to delete `/purchased_discount` objects or retain them with a status of `Canceled`.

Changing the status of a discount offer to `canceled` sets the purchase, usage, and cycle end times to the cancellation time.

When you cancel a discount offer for an account or service, you also cancel the discount offer for each discount sharing group that the account or service owns. For information about discount sharing groups, see "About Charge and Discount Sharing Groups" in *BRM Managing Accounts Receivable*.

See these related topics:

- Configuring Discount End Dates during Mid-Cycle Cancellations
- Rating Delayed Events for a Canceled Discount Offer
- Changing the Status of Discounts Canceled in Mid-Cycle
- Deleting Canceled Discount Offers

### Configuring Discount End Dates during Mid-Cycle Cancellations

When a discount is canceled in the middle of an accounting cycle and the proration for the discount is set to **Full discount**, you can configure two ways to handle the discount:

- By default, BRM sets the discount end date to the end date of the accounting cycle.
- You can configure BRM to cancel the discount immediately. In this case, BRM sets the discount end date to the cancellation date.

To enable this feature, run the `pin_bus_params` utility to change the `CancelFullDiscountImmediate` business parameter. For information about this utility, see *BRM Developer's Guide*.

To immediately cancel discounts canceled in mid-cycle:

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
   <CancelFullDiscountImmediate>enabled</CancelFullDiscountImmediate>
   ```
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.

Rating Delayed Events for a Canceled Discount Offer

By default, when you cancel a discount offer instance, you set the /purchased_discount object's status to Canceled. When you set up BRM to use delayed billing, BRM requires information about the canceled discount offers. BRM uses this information to rate the delayed events that were generated before the discount offer was canceled. To rate events for canceled discount offers, the canceled discount offers must not be deleted.

BRM does not delete canceled discount offers by default. To ensure that canceled discount offer objects are not deleted, do the following:

1. Open the CM configuration file (BRM_home/sys/cm/pin.conf).
2. If necessary, set the value of the keep_cancelled_products_or_discounts entry to 1.
3. Save and close the file.
4. Restart the CM.
5. Edit the PCM_OP_SUBSCRIPTION_POL_SPEC_CANCEL_DISCOUNT policy opcode. By default, this policy opcode keeps/purchased_discount objects when they are canceled. Verify that you set the opcode to cancel, not delete, the discount offer.

When you change the status of a discount offer to Canceled, you also set its purchase, usage, and cycle end dates to the cancellation date.

Changing the Status of Discounts Canceled in Mid-Cycle

When an entire discount is applied to discounts that are canceled in the middle of a cycle, BRM sets the discount to expire at the end of the cycle, but its status remains active.

To change the status of expired discounts from active to canceled, you must run the pin_discount_cleanup utility with the -m parameter.

You can run this utility daily or add it to the pin_bill_day utility to be run automatically.

Deleting Canceled Discount Offers

You might want to delete canceled discount offers from your database (for example, to improve performance by reducing the amount of data stored in the database).
To delete `/purchased_discount` objects when you cancel a discount offer:

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the value of the `keep_cancelled_products_or_discounts` entry.
   - 0 deletes `/purchased_discount` objects when you cancel the discount offer.
   - 1 (default) keeps the canceled discount offers in `/purchased_discount` objects.
3. Save and close the file.
4. Restart the CM.
Managing Purchased Bundles

This document describes how to manage Oracle Communications Billing and Revenue Management (BRM) customers’ purchased bundles.

Topics in this document:

• Modifying Bundles
• Configuring Bundle Dependencies
• Canceling Bundles

See also:

• List of Customer Management Features
• BRM Concepts

Modifying Bundles

You use Billing Care or Customer Center to modify bundles owned by customer accounts (for example, to transition to a new bundle).

You can change the valid bundle period or the charge offers associated with the bundle, and you can define prerequisites, mutually exclusive relationships, or transitions for bundles.

Configuring Bundle Dependencies

You can set up bundle dependencies that determine which bundles you allow a customer to own.

To be able to check that no bundle violates the prerequisites and mutual exclusivity rules, you need to enable bundle dependencies validation.

In PDC or Pricing Center, you can define dependencies between bundles that set up the following relationships:

• **Prerequisites.** Specifies that an account must own a particular bundle to be able to purchase an additional bundle.

  A prerequisite can contain bundles of different services. For example, to own a GPRS bundle, an account must own a GSM bundle.

  A prerequisite bundle cannot contain item charge offers. When you create a package with alternate bundles, and if the base bundle contains an item charge offer, the purchase fails.

• **Required bundle.** Specifies whether bundles are optional or required for packages. *Required* bundles must be purchased when a package is purchased, whereas *optional* bundles can be added to an account at any time.
• **Mutual exclusivity.** Sets up a mutually exclusive relationship between two bundles, so if an account owns one bundle, it cannot own the other.

• **Allowed transitions.** Specifies which bundles or packages can serve as replacements for others.

  Transitions specify the bundles that customers can switch to and remain fully provisioned. While transitioning from one bundle to another, your customers retain their devices, such as phone numbers and services.

  Customers owning bundles associated with a primary service may transition to other bundles associated with that primary service. The list of bundles displayed as available for transition are all associated with a specific primary service.

To enable the bundle dependencies validation.

1. Open the CM configuration file `BRM_home/sys/cm/pin.conf`.
2. *Uncomment* the `validate_deal_dependencies` entry to enable the bundle dependencies validation:
   
   ```
   fm_utils validate_deal_dependencies 1
   ```
3. Save and close the file.
4. Stop and restart the CM.

### Bundle Dependency Validations Involving Inactive or Canceled Charge Offers or Discount Offers

By default, BRM does not perform bundle dependency validations for inactive or canceled charge offers or discount offers.

To enforce bundle dependency validations for inactive or canceled charge offers or discount offers, you need to configure BRM to do the following:

• Maintain any deleted charge offers in the `/purchased_product` objects to enable charge offer and discount offer validations.

  To do so, set the value of the `keep_cancelled_products_or_discounts` entry to `1` in the `pin.conf` configuration file for Connection Manager. See "Deleting Canceled Charge Offers" for more information.

• Perform prerequisite and mutual exclusivity rule dependency validation.

  To do so, verify that the `validate_deal_dependencies` entry is uncommented in the `pin.conf` configuration file for Connection Manager.

• Perform bundle dependency validations on inactive or canceled charge offers and discount offers.

  To do so, enable the `ProductLevelValidation` business parameter in the `/config/business_params` object in the `BRM_home/sys/data/config/bus_params_subscription.xml` file.

  By default, `ProductLevelValidation` is set to `disabled`. 
Enabling Bundle Dependency Validations for Inactive or Canceled Offers

When BRM is configured to enforce bundle dependency validations for inactive or canceled charge offers or discount offers and a bundle A is the prerequisite of bundle B:

- If bundle B is owned by the account, BRM will not allow cancellation or inactivation of any charge offers or discount offers of bundle A unless all the charge offers or discount offers of bundle B are in canceled or inactive state.
- If any charge offer or discount offer of bundle A is in the canceled state, bundle B cannot be purchased.
- If any charge offer or discount offer of bundle A is in the inactive state, bundle B can be purchased only in an inactive state; that is, all charge offers and discount offers of bundle B have to be purchased as inactive. All the charge offers or discount offers in prerequisite bundle A should be first activated before activating the charge offers or discount offers in bundle B.

Note:

- BRM does not perform validations between the validity dates of the prerequisite and dependent bundles.

To enable this feature, run the `pin_bus_params` utility to change the `ProductLevelValidation` business parameter. For information about this utility, see `BRM Developer’s Guide`.

To enable bundle dependency validations for inactive or canceled charge offers and discount offers:

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
   <ProductLevelValidation>enabled</ProductLevelValidation>
   ``
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*.

**Canceling Bundles**

You can cancel all the charge offers in a bundle in one operation, or you can cancel individual charge offers. Canceling a bundle is particularly useful when a customer wants to upgrade. You can cancel all of the charge offers in the old bundle as a group before purchasing the new bundle for the customer's account. You can cancel a bundle immediately or backdate the cancellation of a bundle to a date earlier than the current date.

**Note:**

- After you cancel a bundle, you cannot reactivate the bundle or the charge offers it contains.
- You cannot cancel a required bundle. Instead, either transition to a new package or close the service associated with the bundle.
Configuring Bundle and Package Transitions

Transitions determine the conditions that must be met to purchase a bundle or package. This document describes how to manage Oracle Communications Billing and Revenue Management (BRM) bundle and package transitions.

Topics in this document:

• Transitioning Bundles
• Transitioning Packages
• About Defining Package Transition Rules
• Creating Custom Transition Types for Bundles and Packages

See also:

• List of Customer Management Features
• BRM Concepts

Transitioning Bundles

You can set up transitions that determine which bundles can be purchased subsequent to another bundle.

To transition bundles, you use PDC or Pricing Center to set up transition rules, which are applied when you upgrade or downgrade a bundle for a customer. This enables you to limit the bundles that customers can switch to and still remain fully provisioned.

A bundle cannot be transitioned under the following circumstances:

• The new bundle is mutually exclusive to a bundle currently in the account.
• The account is missing a necessary prerequisite bundle.
• The bundle is associated with an inactive service.
• The service being transitioned from and the service being transitioned to are not the same service.
• The bundle is required in the associated package.

If a service was closed because it was associated with a package transition, you cannot change the closed status of the service.

To customize how to transition bundles, use the PCM_OP_SUBSCRIPTION_POL_PRE_TRANSITION_DEAL opcode. See BRM Opcode Guide.
Transitioning Packages

You specify the rules that govern the transition of an account from the source package to a target package. BRM imposes certain limitations on when accounts can transition to or from other packages. It requires you to define package transition rules for each package-to-package transition by manually configuring the transition rules in PDC or Pricing Center.

You can use the PCM_OP_SUBSCRIPTION_TRANSITION_PLAN and PCM_OP_SUBSCRIPTION_POL_PRE_TRANSITION_PLAN opcodes to customize package transitions. See BRM Opcode Guide.

When you transition accounts from one package to another package in BRM:

- If the transition type is a generation change, the two packages do not have to share the same primary service. If the transition type is an upgrade or a downgrade, the two packages must share the same primary service.
- If the source and target packages are a valid combination, you can configure the transition rule such that the source package retains the noncurrency grants as valid to the end of the cycle. To do so, set the value of PIN_FLD_FLAGS input to the PCM_OP_SUBSCRIPTION_TRANSITION_PLAN opcode to be PIN_SUBS_TRANSITION_CONTROL_ROLLOVER. (See pin_subscription.h.)
- BRM checks the status of any add-on bundles (that are not part of the source package) owned by the account. If all add-on bundles are canceled, BRM closes the associated services and transitions the account to the target package. If the add-on bundles are not canceled, BRM returns an error and retains the source package for the account. Therefore, you must first cancel all add-on bundles owned by the account before you transition the account to the target package.

The following restrictions apply to package transitions in BRM:

- You cannot backdate a package-transition or generation change to a prior period.
- If you delete a service from a package, BRM closes that service and sets its status to PIN_STATUS_FLAG_DUE_TO_TRANSITION. You cannot change the status of a closed service.
- BRM does not transfer extended rating attributes (ERAs) data during a package transition or a generation change. For example, if you have an account with ERA on friends and family and perform a package transition or generation change, the ERA data is not transferred to the new package.
- BRM, by default, retains the credit limits associated with the source package for an account when the source and target packages for that account are associated with the same balance group but each package has different credit limits. To set new credit limits for the account, you can customize PCM_OP_CUST_POL_TRANSITION_PLAN opcode by doing the following:
  1. Set the credit limit in the PIN_FLD_LIMITS array.
  2. Pass this array in the input flist to the PCM_OP_SUBSCRIPTION_TRANSITION_PLAN opcode,
You can manually configure the package transition rule for each package-to-package transition in PDC or Pricing Center. For each such package transition rule you configure, BRM creates a /transition object to store the rules.

You can perform package transitions to any package without creating transition objects. This is a useful approach because for each package in the BRM system, you need to specify all other packages to which the package can transition. For example, if you have 300 packages and each package can transition to or from any other package, you define 89,401 (299 x 299) package transition rules, thus creating 89,401 /transition objects.

In addition, every time you add a package, you must define the transition rules for each existing package to point to the new package. As a result, the number of transition rules that you need to define in PDC or Pricing Center increases in proportion to any increase in the number of packages you support.

You can perform package transitions to any package without creating /transition objects to store the transition rules.

You use the PCM_OP_SUBSCRIPTION_POL_PRE_TRANSITION_PLAN policy opcode to automatically enable package transitions to any package without /transition objects. See BRM Opcode Guide.

Defining a Generation Change for Packages

A generation change enables you to transition your customers between 2G (second generation) and 3G (third generation) wireless packages and services. Packages are called 2G or 3G depending on whether their primary service type runs on a 2G or 3G wireless network. A 3G wireless network is faster than a 2G network and can transmit video and two-way video telephone calls.

You can use any 2G or 3G service type, such as the following:

- /service/telco/pdc, a 2G service type based on the Personal Digital Cellular standard for digital mobile telephony.
- /service/telco/imt, a 3G service type based on the International Mobile Telecommunications (IMT) standard for 3G wireless communications.

Whether you are transitioning a customer from a 2G to a 3G package or from a 3G to a 2G package, both packages are valid all day on the day of transition. The package that is being phased out expires at 00:00:00 hours at the end of the transition day; the package being transitioned to becomes valid at 00:00:00 hours at the beginning of the transition day.

You set the transition rules between two packages. You can also set up standalone bundles as add-ons and specify that the transition rules apply to those bundles as well.
When a transition between two packages is under way, no other transition is allowed between the two packages for the duration of the transition day.

You can backdate the subscription actions to a prior period in case of a generation change, but doing so can lead to incorrect results.

For the package replacing the current package, the following happens at 00:00:00 hours at the start of the transition day:

- The purchase, cycle, and usage start dates are set to 00:00:00 hours at the beginning of the transition day.
- All dependent services and bundles associated with the package are included in the transition.
- Any cycle fees for this package are charged from 00:00:00 hours at the beginning of the transition day to the end of the billing cycle.

Purchase and cancel fees can be configured to be waived during the transition.

The following happens at transition time or slightly after:

- All configured bundles are purchased for the service.
- The new service is provisioned.

For the package you are phasing out, the following happens at 00:00:00 hours at the end of the transition day:

- The current service is closed.
- Any dependent services are closed.
- All associated charge offers and dependent services are canceled.
- Forward and arrears cycle fees are prorated from the beginning of the billing cycle to the cancellation time.

Configuring Services for a Generation Change

By default, the service being phased out is active for one day, the day the generation change takes place. You can configure the number of days both services involved in a generation change are active.

To enable this feature, run the `pin_bus_params` utility to change the `ProdEndOffsetpackageTransition` business parameter. For information about this utility, see BRM Developer's Guide.

To modify the number of days the phased-out service is active:
1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```bash
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
3. In the file, change `10` to the number of days you want the phased-out service to remain active (the default is 10 days). For example, if you change the value to `15`, the phased-out service remains active 15 days after generation of the new service. The minimum number of days you can keep a phased-out service active is `1` and the maximum is `31`:
   ```xml
   <ProdEndOffsetpackageTransition>10</ProdEndOffsetpackageTransition>
   ```
4. Save the file as `bus_params_billing.xml`.
5. Load the XML file into the BRM database:
   ```bash
   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*.

### Creating Custom Transition Types for Bundles and Packages

By default, BRM provides these transition types for packages and bundles:

- Upgrade
- Downgrade
- Generation Change

To add custom transition types for bundles and packages:

1. Open the `BRM_home/sys/data/pricing/example/pin_transition_type` file in a text editor.
2. Add your custom transition types by using this syntax:
   ```plaintext
   TransitionIDNumber    TransitionString
   ```
   where:
   - TransitionIDNumber specifies the ID of the transition type. This value will be visible in the `transition` object. Your custom ID numbers must be unique and greater than the number `100`. However, they need not be in numerical order.
   - TransitionString specifies the transition name that is displayed in PDC or Pricing Center.

For example, add the following line to create a custom transition type named RED:

   ```plaintext
   101  RED
   ```
3. Save and close the file.
4. Go to the directory in which you saved the `pin_transition_type` file and enter the following command:
   ```bash
   load_transition_type [TransitionTypeFile]
   ```
where TransitionTypeFile specifies the name and location of the file that contains your custom transition types. By default, the utility uses the `pin_transition_type` file in the directory from which you run the utility.

![Note:]

For more information about the utility's syntax, see "\texttt{load_transition_type}".

5. Restart PDC or Pricing Center.

Your new transition types are loaded into the `\texttt{config/transition_type}` object and are displayed in PDC or Pricing Center the next time you start it.
Backdating Subscription Actions

This document describes how to backdate subscription actions in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:
- About Backdating Subscription Actions
- About Backdating Beyond the G/L Posting Date
- How Effective Time Is Used to Validate Backdating Operations
- Rerating of Events for the Backdated Period
- About Backdated Charge Offer, Discount Offer, or Bundle Cancellation

See also:
- List of Customer Management Features
- BRM Concepts

About Backdating Subscription Actions

Subscription actions backdating lets you perform certain subscription actions so that the operations take effect on a prior date instead of the date it occurred. You can perform subscription backdating operations to rectify errors committed while performing subscription operations.

For example, you might want to backdate a subscription operation when:

- A customer requests a charge offer cancellation on October 30, but the cancellation is not recorded in the BRM system until November 15. The customer is charged for the period of October 30 to November 15.
- A customer requests a charge offer purchase on January 1, but the purchase date is entered in the BRM system as February 1.

BRM supports the following subscription backdating operations:

- Backdated account creation.
- Backdated account and service status change. See "About Backdated Status Change".
- Backdated charge offer, discount offer, or bundle cancellation. See "About Backdated Charge Offer, Discount Offer, or Bundle Cancellation".
- Backdated charge offer or discount offer purchase, usage, or cycle start and end date.
- Backdated remaining charge discount offer purchase, cancellation, and modification if the proration options are set to Prorated discount. See the discussion of proration options in BRM Creating Product Offerings.

You can perform subscription backdating by using one of the following methods:
• By using Billing Care.
• By using Customer Center.
• By passing the backdated date in the PIN_FLD_END_T field of the corresponding subscription opcodes.

To backdate a charge offer, discount offer, bundle, or package purchase, use the PCM_OP_SUBSCRIPTION_PURCHASE_PRODUCT opcode. See BRM Opcode Guide.

About Backdated Status Change

BRM supports backdating the status change of a charge offer, discount offer, service, or account to a prior date.

BRM does not allow backdating the status change in the following situations:

• The backdated date is prior to the G/L posting date. See "About Backdating Beyond the G/L Posting Date”.
• The backdated date is prior to the effective date of the account, service, charge offer, or discount offer. See "How Effective Time Is Used to Validate Backdating Operations”.
• The backdated date is prior to the date of the last status change. You can backdate any status change only up to the date the last status change happened; undoing previous status changes is not allowed.

For example, if you change the status of an account from active to inactive on September 1, you cannot backdate a status change on October 1 to a date prior to September 1.

How Cycle Fees are Calculated for Backdated Status Change

When you backdate the status change, a charge or a refund is applied for the backdated period.

For example, consider that on July 1, you backdate the status of an active account to make it inactive effective from June 1. If the cycle fee is already applied for the period of June 1 to July 1, another cycle fee event is generated that calculates and refunds the charges.

You must rerate any usage events that have occurred in the backdated period to account for the status change.

About Backdating Beyond the G/L Posting Date

BRM does not allow backdating beyond the G/L posting date. When you post a G/L report, you prevent backdating adjustments, write-offs, or subscription transactions, prior to the last date you posted the G/L report. This maintains general ledger data integrity. After they are posted, the G/L report updates the /data/ledger_report object. This object records the date of the latest posted G/L report and controls transaction backdating.

You can run G/L reports at any time without posting data. When you do not post data, backdating is not affected.
How Effective Time Is Used to Validate Backdating Operations

BRM does not allow backdating if the date to which you want to backdate the operation on an account, service, charge offer, or discount offer is prior to the respective effective dates (creation dates).

Rerating of Events for the Backdated Period

Backdated purchase or cancellation of a charge offer or discount offer automatically triggers rerating. In these cases, BRM uses event notification to create rerate jobs that rerate the events in the charge offer or discount offer. In all other subscription backdating scenarios, you must either run rerating manually or configure BRM to created rerate jobs automatically.

For example:
- You might manually need to rerate the usage events that had been rated prior to the backdating action.
  For example, on October 15, a usage event for a charge offer is charged at $2.00 per minute. On November 1, a new charge offer is purchased backdated to September 1. The new charge offer has usage pricing of $1.00 per minute and has a higher priority than the first charge offer. In order for the usage that occurred on October 15 to get rated with the new charge offer, you must rerate the usage events manually.
- If you backdate the cancellation of a shared discount offer, you will need to rerate the accounts or services in the group, because the accounts or services might have used the discount offer.

You can configure BRM to rerate backdated operations. See BRM Implementing Charging.

About Backdated Charge Offer, Discount Offer, or Bundle Cancellation

The cancellation of a charge offer can be backdated as far back as the purchase date of the charge offer.

Note:
If you backdate a discount offer purchase to a date prior to the charge offer purchase date, the discount will not be applied in the bill of the current cycle. The discount will be applied in the bill of the next cycle.
Note:

If you backdate the purchase of a discount offer on a cycle forward or cycle arrears event to a previous accounting cycle, the discount offer will be refunded only for the current accounting cycle.

BRM does not allow you to backdate the charge offer, discount offer, or bundle cancellation if:

- The backdated cancellation date is prior to the G/L posting date. See "About Backdating Beyond the G/L Posting Date".

- The backdated cancellation date is prior to the charge offer or discount offer's effective date. For example, a charge offer or discount offer that was purchased to be effective from January 15 cannot have any event on it backdated prior to January 15. See "How Effective Time Is Used to Validate Backdating Operations".

- The backdated cancellation date is prior to the date of the last status change of the account or service.

For example, consider that you create an account on December 1. On December 10, you change the status of the account to Inactive. On December 15, you change the status of the account back to Active. You cannot backdate a charge offer cancellation prior to December 15, which is the date of the last status change.

Note:

- When you backdate the cancellation of a fold, rollover charge offer, or billing time discount, you must run rerating to apply the corrections on the events that occurred after the backdated date.

- Backdating discounts with discount offer proration options is supported only for discounts that are prorated.
Creating Custom Service Life Cycles

Custom life cycles enable you to manage service status at a detailed level. For example, you can use statuses such as Dormant, Credit Expired, and Fraud Investigated to indicate the exact state of a service and control how the service is used.

This document describes how to configure and use custom service life cycles in Oracle Communications Billing and Revenue Management (BRM).

See also:
- Managing Account and Service Status
- List of Customer Management Features
- BRM Concepts

About Custom Service Life Cycles

By default, all BRM service types use the following statuses: Active, Inactive, and Closed. If you need a life cycle that better represents the phases of a particular service type, create a custom service life cycle for that service type. For example, you might create the following custom life-cycle states:

- Preactive to indicate a service that is ready to activate, but not yet activated.
- Fraud Investigated to indicate a service that is under investigation.
- Dormant to indicate a service that hasn't been used for a specified amount of time.

Custom service life cycles can contain any number of states and state transitions. For each state, you can specify the following:

- The states to which the state can change
- The actions that occur before and after a state transition takes place
- Rules to validate requests received in the state

You can associate custom life cycles with any BRM service type.

To create custom life cycles, you edit and load the config_lifecycle_states.xml file.

Note:
The default service life cycle status is stored in the PIN_FLD_STATUS field in /service objects. The custom service life cycle state is stored in the PIN_FLD_LIFECYCLE_STATE field of /service objects.
Triggering State Changes in Custom Service Life Cycles

The state of a service changes as the result of actions or conditions that affect the service. The actions can be triggered manually or automatically.

In custom service life cycles, state changes can be triggered by the following:

- **CSRs**: Using Billing Care or Customer Center, a customer service representative (CSR) can manually perform any permitted state change. Such changes are defined in the `<TRANSITIONS>` child element of the `<STATES>` element in the `config_lifecycle_states.xml` file.

- **Any action that impacts a service balance**: After a balance is adjusted or topped up, the PCM_OP_BAL_POL_CHECK_LIFECYCLE_STATE policy opcode triggers any required service state changes and updates the state expiration date (PIN_FLD_SERVICE_STATE_EXPIRATION_T field) in the `/service` object based on the new state’s expiration period.

  If you create your own custom service life cycles, you can modify this policy opcode to trigger state changes based on different criteria.

- **pin_state_change**: This utility performs bulk service state changes based on the state expiration time (PIN_FLD_SERVICE_STATE_EXPIRATION_T field) in `/service` objects).

  A system administrator schedules the utility to run at a specified time each day. When the utility finds services whose state expires on the current date, it changes that state to its default next state.

  _Note:_ The default next state is specified in the `<TRANSITIONS>` element whose `<DEFAULT_FLAG>` is set to 1. If this flag is not set to 1 in any of the transitions configured for a state, this utility does not change the state.

  See "pin_state_change" for more information.

Configuring Business Rules for Custom Service Life Cycles

For each life-cycle state, you can configure business rules to validate the actions that subscribers try to perform in the state.

You configure business rules in the `<RULES>` child element of the `<STATES>` element in the `config_lifecycle_states.xml` file. In that file, each business rule includes the following elements:

- **<MODULE>**: The name of the facilities module (FM) that uses the rule. (The rule is coded in the FM.)

- **<NAME>**: The name of the rule as it appears in the FM.

- **<VALUE>**: A value that indicates whether the business rule is enabled or disabled for the state.
To create and use a custom business rule:

1. Code the validation logic for the new rule in the appropriate policy opcode.
2. Include a name for the rule in the policy opcode.
3. Configure a `<RULES>` element for the rule in the appropriate service life cycle state.

About Creating Custom Service Life Cycles

In addition to defining and loading custom life cycles, you need to do the following:

1. Enable customer service life cycles in BRM. See "Enabling BRM to Use Custom Service Life Cycles".
2. Add SLM entries to the Connection Manager (CM) `pin.conf` file. See "Adding SLM Entries to the CM pin.conf File".
3. Map each state in the custom service life cycle to a status in the default service life cycle. See "Mapping States to Statuses".
4. Associate one or more services with the custom service life cycle. See "Associating Services with Custom Life Cycles".
5. Associate account bill units (`/billinfo` objects) with the SLM business profile. See "Associating Bill Units with the SLM Business Profile".

Creating Custom Service Life Cycles

To create a custom service life cycle:

1. Open the `config_lifecycle_states.xml` file in an XML editor or a text editor. By default, the file is in the `BRM_home/syndata/config` directory.

   ![Note](image)

   The `config_lifecycle_states.xml` file can contain multiple service life cycle configurations. In the XML file, each life cycle configuration is identified by its `<NAME>` element. When the content of the XML file is loaded into the BRM database, each life cycle configuration is put into a separate `/config/lifecycle_states` object.

2. Add a `<ConfigObject>` element to the `<ObjectList>` element.
3. In the `<ConfigObject>` element, specify values for the life cycle elements.
4. Save and close the file.
5. Open the `pin.conf` file for the `load_config` utility in `BRM_home/apps/load_config`.
6. Add the following line to enable validating the customer life cycles:

   ```
   load_config validation_module libLoadValidSLM LoadValidSLM_init
   ```

7. Save and close the `pin.conf` file.
8. Run the following command, which loads the `config_lifecycle_states.xml` file:
9. Stop and restart the CM.

Modifying Custom Service Life Cycles

To modify a custom service life cycle:

1. Open the config_lifecycle_states.xml file in an XML editor or a text editor. By default, the file is in the BRM_home/sys/data/config directory.

2. Modify the <ConfigObject> element in which the life cycle is configured.

Note:

- If you change the <NAME> element of a life cycle that services are using, you must update the value of the lifecycle_obj key in those services' validation templates in the pin_slm_business_profile.xml file. You must then reload that file.

- For details about editing and loading pin_slm_business_profile.xml, see “Associating Services with Custom Life Cycles”.

3. Set the configMode attribute of the <ObjectList> element to one of the following values:
   - replace: (Default) Updates the existing /config/lifecycle_states objects.
   - recreate: Deletes the existing /config/lifecycle_states objects and creates new objects.

4. Save and close the file.

5. Run the following command, which loads the modified config_lifecycle_states.xml file:

   load_config config_lifecycle_states.xml
### Note:

- The "load_config" utility needs a configuration (pin.conf) file in the directory from which you run the utility.
- The pin.conf file must contain the following entry:
  
  ```bash
  - load_config validation_module libLoadValidSLM LoadValidSLM_init
  ```

  This entry enables the utility to validate the XML file values.
- If you do not run the utility from the directory in which the configuration file is located, include the complete path to the file. For example:

  ```bash
  load_config BRM_home/sys/data/config/config_lifecycle_states.xml
  ```

The utility converts the XML file into one or more /config/lifecycle_states objects, depending on the number of &lt;ConfigObject&gt; elements in the XML file. Each object contains the life cycle defined in one &lt;ConfigObject&gt; element.

6. Stop and restart the CM.

### Enabling BRM to Use Custom Service Life Cycles

To use custom service life cycles, you must run the `pin_bus_params` utility to change the `SubscriberLifeCycle` business parameter. For information about this utility, see `BRM Developer's Guide`.

To enable BRM to use custom service life cycles:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `lifecycle_params` object:

   ```bash
   pin_bus_params -r BusParamsCustomer bus_params_customer.xm
   ```

3. In the file, change `disabled` to `enabled`:

   ```xml
   &lt;SubscriberLifeCycle&gt;enabled&lt;/SubscriberLifeCycle&gt;
   ```

4. Save the file as `bus_params_customer.xml`.
5. Load the XML file into the BRM database:

   ```bash
   pin_bus_params bus_params_customer.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`.

### Adding SLM Entries to the CM pin.conf File

To support custom service life cycles, BRM needs the following entries in the CM `pin.conf` file:
If these entries are not in your CM pin.conf file, you must add them.

To add the CM pin.conf entries for custom service life cycles:

1. Open the CM pin.conf file in BRM_home/sys/cm.
2. Add the following entries to the pin.conf file:
   
   - cm_cache fm_bill_utils_business_profile_cache number_of_entries, cache_size, hash_size
   - cm_cache fm_bill_template_cache number_of_entries, cache_size, hash_size
   - cm_cache fm_cust_lifecycle_config_cache number_of_entries, cache_size, hash_size
   - cm_cache fm_cust_statemap_config_cache number_of_entries, cache_size, hash_size

   Where
   
   - number_of_entries is the following:
     
     - For fm_bill_utils_business_profile_cache, the total number of business profiles (config/business_profile objects), including service life cycle business profiles, you plan to create in your system. See "Associating Services with Custom Life Cycles" for more information.
     
     - For fm_bill_template_cache, the total number of service validation templates (config/template/service objects), including those defined in service life cycle business profiles, you plan to create in your system. See "Associating Services with Custom Life Cycles" for more information.
     
     - For fm_cust_lifecycle_config_cache, the number of config/lifecycle_states objects you plan to create in your system. Each object represents one custom service life cycle. See "Creating Custom Service Life Cycles" for more information.
     
     - For fm_cust_statemap_config_cache, the number of config/service_state_map objects you plan to create in your system. The only valid value is 1. See "Mapping States to Statuses" for more information.
   
   - cache_size is the sum of the sizes of the cached objects in bytes.
   
   - hash_size is the square root of number_of_entries.

3. Save and close the file.
4. Stop and restart the CM.

Mapping States to Statuses

Each service life cycle state must be mapped to a status (Active, Inactive, or Closed) in the default service life cycle. This mapping supports the following:

- Backward compatibility
- Custom service state changes triggered by account status changes
• Member service state changes triggered by subscription service status changes

**Note:**

You should not include both services that use the default service life cycle and services that use a custom service life cycle in a service group. If you do, the subscription service might be inactive while a member service is active.

A state can be mapped to only one status.

A status can be mapped to multiple states.

To create state-to-status mapping for the states in all your custom service life cycles, edit the service state mapping configuration file (`BRM_home/sys/data/config/config_service_state_map.xml`). See "About the Service State Mapping Configuration File" for more information.

After editing the configuration file, use the `load_config` utility to load the file's contents into the `/config/service_state_map` object in the BRM database. That object contains the state-to-status mapping for every custom service life cycle in your system. See "Mapping States to Statuses" for more information.

**About the Default State of a Status**

Consider the following situation:

• BRM needs a state for a service that uses a custom life cycle.

• Only the service status is known.

• The status is mapped to multiple states.

In this situation, BRM uses the status’s default state, which is the state whose PIN_FLD_DEFAULT_FLAG field is set to 1 for that status in the `/config/service_state_map` object.

For example, suppose the Active (10100) status is mapped to the Active, Recharge Only, and Credit Expired states. If a state value is required for a service whose status (Active) is known but whose state is unknown, BRM uses the state in the PIN_FLD_STATES array of the `/config/service_state_map` object that contains the following values:

• PIN_FLD_DEFAULT_FLAG = 1

• PIN_FLD_STATUS = 10100

**Note:**

When a status is mapped to multiple states, the default flag of only one of those states can be set to 1.
Every service life cycle state defined in your system must have a transition to each status's default state. This enables BRM to complete a state transition when only the status to which the service must change is known.

For example, if BRM receives a request to change a status from Active to Inactive and the current state of the service is Recharge Only, a transition from Recharge Only to Suspended (the default state for the Inactive status) must exist. If the transition is not defined, the transaction fails.

State transitions are defined in the `config_lifecycle_states.xml` file. See "Creating Custom Service Life Cycles" for more information.

### About the Service State Mapping Configuration File

You configure the state-to-status mapping for the states in all your custom service life cycles in the `BRM_home/sys/data/config/config_service_state_map.xml` file.

The mapping for each service life cycle state in your BRM system is configured in a `<States>` element, which has the following syntax:

```
<ObjectList>
  <ConfigObject>
    <DESCR>Description</DESCR>
    <NAME>Name</NAME>
    <STATES elem="0">
      <DEFAULT_FLAG>DefaultFlag</DEFAULT_FLAG>
      <LIFECYCLE_STATE>LifeCycleStateNumber</LIFECYCLE_STATE>
      <STATUS>StatusNumber</STATUS>
      <STATUS_FLAGS>StatusFlagNumber</STATUS_FLAGS>
    </STATES>
  </ConfigObject>
</ObjectList>
```

Table 17-1 lists the elements of the preceding syntax:

<table>
<thead>
<tr>
<th>XML Element</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPT</td>
<td>(Optional) Character string that describes the state-to-status mapping.</td>
<td>Minimum length is 0 characters. Maximum length is 255 characters.</td>
</tr>
<tr>
<td>NAME</td>
<td>Character string used as the name of the mapping.</td>
<td>The name must be unique within your BRM system. Minimum length is 1 character. Maximum length is 255 characters.</td>
</tr>
</tbody>
</table>
Table 17-1  (Cont.) Service State Mapping Elements

<table>
<thead>
<tr>
<th>XML Element</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATES</td>
<td>Parent element of the child elements that map a state to a status. This element is mapped to the PIN_FLD_STATES array in the /config/service_state_map object. The array must contain all the the service life cycle states in your system. It can contain states from multiple service life cycles. See “Mapping States to Statuses”.</td>
<td>If you configure multiple &lt;STATES&gt; elements, the integer value of each element's elem attribute must be different. The values do not have to be sequential, and they have no relationship to the values of the elem attributes of other elements.</td>
</tr>
</tbody>
</table>
| DEFAULT_FLAG   | Integer that specifies whether this is the default state for the status to which it is mapped. A status can be mapped to multiple states, but only one of those states can be the status's default state. If a state is required when only a status value is available, BRM uses the default state for that status. | Specify one of the following values:  
  • 1: Default state. If multiple states are mapped to a status, only one of those states can have this value.  
  • 0: Not the default state. |
| LIFECYCLE_STATE| Integer that specifies the numeric ID of a custom service state. Specifying the unique ID of the custom service state to which this mapping applies. This ID is configured in the config_lifecycle_states.xml file. | Specify the unique ID of the custom service state to which this mapping applies. This ID is configured in the config_lifecycle_states.xml file. |
| STATUS         | Integer that specifies the numeric ID of the status in the default service life cycle to which this mapping applies. | Specify one of the following status IDs:  
  • 10100: Active  
  • 10102: Inactive  
  • 10103: Closed |
| STATUS_FLAGS   | Integer that specifies the status flag to pass when a state change occurs. The status flag specifies the reason for the change. | When a service is reactivated, the value must match the value used in the previous state change.  
  Note: STATUS_FLAGS values are listed in the BRM_home/include/pin_cust.h file. |

Mapping States to Statuses

Note:
A custom life cycle state should not be mapped to more than one status. Each status, however, can be mapped to multiple states.

To create state-to-status mapping:

1. Open the config_service_state_map.xml file in an XML editor or a text editor. By default, the file is in the BRM_home/sys/data/config directory.
2. Add a <States> element to the <ConfigObject> element.
3. In the <ConfigObject> element, specify values for the elements listed in Table 17-1.
4. Save and close the file.

5. Run the following command, which loads the changes into the `config/service_state_map` object in the BRM database:

   ```bash
   load_config config_service_state_map.xml
   ```

   **Note:**
   - The "load_config" utility needs a configuration (`pin.conf`) file in the directory from which you run the utility.
   - The `pin.conf` file must contain the following entry:
     ```
     - load_config validation_module libLoadValidSLM LoadValidSLM_init
     ```
     This entry enables the utility to validate the XML file values.
   - If you do not run the utility from the directory in which the configuration file is located, include the complete path to the file. For example:
     ```bash
     load_config BRM_home/sys/data/config/config_service_state_map.xml
     ```
   The utility converts the XML file into one `config/service_state_map` object.

6. Read the updated object with the `testnap` utility's `robj` command or with Object Browser to verify that all fields are correct.

   For information about reading an object and writing its contents to a file, see "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.

7. Stop and restart the CM.

### Modifying State-to-Status Mapping

To modify state-to-status mapping:

1. Open the `config_service_state_map.xml` file in an XML editor or a text editor.
   By default, the file is in the `BRM_home/sys/data/config` directory.

2. Modify the `<States>` element in which the mapping is configured.
   See Table 17-1 for more information about the elements of state-to-status mapping.

3. Set the `configMode` attribute of the `<ObjectList>` element to one of the following values:
   - `replace`: (Default) Updates the existing `config/service_state_map` objects.
   - `recreate`: Deletes the existing `config/service_state_map` objects and creates new objects.

4. Save and close the file.
5. Run the following command, which loads the changes into the `config/service_state_map` object in the BRM database:

```
load_config config_service_state_map.xml
```

**Note:**

- The "load_config" utility needs a configuration (`pin.conf`) file in the directory from which you run the utility.
- The `pin.conf` file must contain the following entry:
  ```
  - load_config validation_module libLoadValidSLM LoadValidSLM_init
  ```
  This entry enables the utility to validate the XML file values.
- If you do not run the utility from the directory in which the configuration file is located, include the complete path to the file. For example:
  ```
  load_config BRM_home/sys/data/config/config_service_state_map.xml
  ```
  The utility converts the XML file into one `config/service_state_map` object.

6. Read the updated object with the `testnap` utility's `robj` command or with Object Browser to verify that all fields are correct.

   For information about reading an object and writing its contents to a file, see "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.

7. Stop and restart the CM.

### Deleting State-to-Status Mapping

To delete the state-to-status mapping from your BRM system, use the following `load_config` command:

```
load_config -r service_state_map
```

**WARNING:**

Do not delete mapping that a service is using. Doing so will cause data corruption.

This command deletes the `config/service_state_map` object from your BRM database.

See "load_config" for more information.
Associating Services with Custom Life Cycles

A service type (/service subclass object) can be associated with only one life cycle. By default, all service types use the default service life cycle.

To associate a service type with a custom service life cycle, you add the following key-value pair to that service’s validation template in the SLM business profile:

- **Key:** lifecycle_obj
- **Value:** Name of the custom service life cycle

The SLM business profile is configured in the `pin_slm_business_profile.xml` file.

⚠️ **WARNING:**

After a service type is in use in your BRM system, do not associate it with a different life cycle. If you do, state and status changes might fail and data might be corrupted.

See "Associating Services with Custom Life Cycles" for information about editing the SLM business profile and loading it into the BRM database.


To associate a service type with a custom life cycle:

1. Open the `pin_slm_business_profile.xml` file in an XML editor or a text editor.
   - By default, the file is in the `BRM_home/sys/data/config` directory.
2. In the list of validation template IDs (`<TemplateID>`) elements, add or delete the name and type of the validation template for the appropriate service.
   - By default, this business profile contains these validation template IDs:
     ```xml
     <TemplateId name="ServiceTelcoGprs" type="/service/telco/gprs" />
     <TemplateId name="ServiceTelcoGsm" type="/service/telco/gsm" />
     <TemplateId name="ServiceTelcoGsmTel" type="/service/telco/gsm/telephony" />
     ```
   - See "Defining Business Profiles" for more information about `<TemplateID>` elements.
3. In the list of key values (`<NameValue>`) elements, add or delete key-value pairs to identify the bill units that belong to this business profile.
   - See "Defining Business Profiles" for more information about `<NameValue>` elements.
4. In the list of validation templates (`<TemplateList> parent element), add or delete the definition of the validation template (`<Template> element) for the appropriate service.
5. Save and close the file.
6. Create a `/config/template/service` subclass for each service type in the list of validation template IDs (`<TemplateID>` elements) of the `slm_business_profile.xml` file.

For example, to support the SLM business profile, create the following subclasses:

- `/config/template/service/telco/gprs`
- `/config/template/service/telco/gsm`
- `/config/template/service/telco/gsm/telephony`

7. Run the following command, which loads the SLM business profile into a `/config/business_profile` object in the BRM database:

```
load_pin_business_profile pin_slm_business_profile.xml
```

> **Note:**

- The `load_pin_business_profile` utility needs a configuration (pin.conf) file in the directory from which you run the utility.
- If you do not run the utility from the directory in which the XML file is located, include the complete path to the file. For example:

```
load_pin_business_profile BRM_home/sys/data/config/pin_slm_business_profile.xml
```

See "load_pin_business_profile" for more information.

The `PIN_FLD_NAME` field in the `/config/business_profile` object containing the SLM business profile is set to `SLM`.

8. Read the updated object with the `testnap` utility's `robj` command or with Object Browser to verify that all fields are correct.

For information about reading an object and writing its contents to a file, see "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.

9. Stop and restart the CM.

10. (Optional) Make the SLM business profile your system's default business profile. See "Associating Bill Units with the SLM Business Profile".

**Associating Bill Units with the SLM Business Profile**

For an account to own a service that uses a custom life cycle, the account's bill unit must be associated with the business profile in which the service is linked to the custom life cycle (see "Associating Services with Custom Life Cycles").

By default, services are linked to custom life cycles in the SLM business profile. You can associate an account's bill unit with the SLM business profile in either of the following ways:

- Make the SLM business profile your system's default business profile. In this case, when an account is created, its bill unit is automatically associated with the default business profile. To enable this feature, run the `pin_bus_params` utility to change
the DefaultBusinessProfile business parameter. For information about this utility, see *BRM Developer's Guide*.

- If the SLM business profile is not the system's default business profile, you can associate the SLM business profile with the bill unit after the account is created. See "Overriding the Default Business Profile" in *BRM System Administrator's Guide* for more information.

To make the SLM business profile your system's default business profile:

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 this element value to SLM (the name of the business profile configured in (pin_slm_business_profile.xml):
   ```
   <DefaultBusinessProfile>SLM</DefaultBusinessProfile>
   ```
4. Save the file as bus_params_billing.xml.
5. If the name of your service life cycle business profile is not SLM, do the following:
   a. Open the bus_params_billing.xsl file in an XML editor or a text editor. By default, the file is in the BRM_home/xsd directory.
   b. Search the file for the following section:
      ```
      <xsl:template match="bc:DefaultBusinessProfile">
      ```
   c. In the following lines of that section, replace SLM with the name of your service life cycle business profile:
      ```
      <xsl:when test="normalize-space(text()) = 'SLM'">
      <xsl:text>SLM</xsl:text>
      ```
      For example, if the profile is named XYZ, the lines should look like this:
      ```
      <xsl:when test="normalize-space(text()) = 'XYZ'">
      <xsl:text>XYZ</xsl:text>
      ```
   d. Close and save the file.
6. Load the XML file into the BRM database:
   ```
   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*. 
Creating and Managing Service Groups

Service groups enable customers to purchase, transfer, inactivate, and cancel services as a group; and to share charges, discounts, and promotions among a group of subscribers.

This document describes Oracle Communications Billing and Revenue Management (BRM) service group management.

See also:

• List of Customer Management Features
• BRM Concepts

About Grouping Services by Subscription

In BRM, you can combine services into service groups, which are sets of services associated with a subscription, such as telephony and text messaging services associated with a wireless connection.

A service group consists of a subscription service and any number of member services. 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.

When services are grouped by subscription, you can do the following:

• Enable customers to purchase, transfer, inactivate, and cancel services as a group.
• Apply discounts and promotions owned by the subscription service to the member services.
• Enable member services to benefit from the subscription service's membership in sharing groups.
• Associate member services with the devices owned by the subscription service.
• Keep separate balances for the service group.
• Create a separate bill and invoice for the subscription.

For example, a service group might consist of the services shown in Figure 18-1:
Figure 18-1  Example Service Group

<table>
<thead>
<tr>
<th>Service Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription service: <strong>Line A</strong></td>
</tr>
<tr>
<td>Member services:</td>
</tr>
<tr>
<td>– Telephone service</td>
</tr>
<tr>
<td>– Short message service</td>
</tr>
<tr>
<td>– Data service</td>
</tr>
</tbody>
</table>

**About Service Groups**

In service groups, subscription services and member services as follows:

- **Sharing Charge and Discount Offers**
- **Sharing Balance Groups**

**Sharing Charge and Discount Offers**

Subscription and member services share charge offers as follows:

- If the subscription service owns a charge offer but a member service does not, the member service uses the subscription service’s charge offer to rate its balance impacts.
- If both the subscription service and a member service own charge offers, the member service uses its own charge offer. It cannot use the subscription service’s charge offer.

Subscription and member services share discount offers as follows:

- If the subscription service owns a discount offer but a member service does not, the member service uses the subscription service’s discount offer to rate its balance impacts.
- If both the subscription service and a member service own discount offers, the member service uses both its discount offer and the subscription service’s discount offer.

**Sharing Balance Groups**

A balance group is a collection of sub-balances that track what customers owe or are credited with for one or more services. Services can use the default account balance group, or they can use a service balance group. See *BRM Managing Accounts Receivable* for more information.

If a subscription service has its own balance group:

- All member services that do not have a balance group use the subscription service’s balance group.
- Any member service that has a balance group uses its own balance group instead of the subscription service’s balance group.
About Using Charge and Discount Offers to Rate Service Group Usage

Both subscription services and member services can own charge and discount offers:

- When the subscription service owns charge and discount offers, the offers are used to rate member service usage.
- When the subscription service and a member service own charge offers, the charge offers’ priority determines whether the subscription or member charge offer is used for rating.
- When the subscription service and a member service own discount offers, the discount offers’ priority determines the order in which the discounts are applied.
- If a member service owns a discount offer but does not have its own balance group, its discount is applied to the subscription service’s balance group.

You specify the priority of charge and discount offers when you create them in PDC or Pricing Center.

About Distributing Discounts to Member Services

Discounts owned by a subscription service can be distributed to member services as follows:

- **Through a discount sharing group**: When the subscription service and a member service have separate balance groups, the subscription service can share its discounts with the member service through a discount sharing group. The discount balance impact can be applied to a balance in the subscription service's balance group or in the member service's balance group. See “About Sharing Charges and Discounts in Service Groups”.

- **Through inheritance**: When a member service shares the subscription service's balance group, the member service automatically inherits any discount purchased by the subscription service. The discount balance impact is applied to the subscription service's shared balance group.

**Note:**

When a billing-time discount grants a percentage off subscription fees based on the total usage of all services in the group, the total discount granted can be greater than intended. If you do not want billing-time discounts to be inherited, do one of the following:

- Purchase them for member services instead of for the subscription service.
- Configure BRM to disallow inheritance for billing-time discounts. See “Preventing Member Services from Inheriting Billing-Time Discounts”.
Preventing Member Services from Inheriting Billing-Time Discounts

By default, billing-time discounts owned by a subscription service in a service group are automatically inherited by all member services that share the subscription service's balance group. When the discount is applied at billing time, it is applied individually to each member service.

If a subscription service owns a billing-time discount that grants a percentage off subscription fees, the total discount can be greater than intended when the discount is inherited. For example, a service group includes a subscription service and ten member services that inherit a billing-time discount. The discount provides .05% off the group's aggregated charges. If the aggregated charges total $1000 when billing is run, the discount grants $5 off to each member service that inherits the discount, and $5 off to the subscription service. The total discount is $55, which is an actual discount of .055% off the aggregated charges.

To prevent member services from inheriting billing-time discounts:

1. Open the CM `pin.conf` file in `BRM_home/sys/cm`, where `BRM_home` is the directory where you installed BRM components.

2. Search for the following line:
   
   `- fm_subscription btd_inheritance = 1`

3. Set the value of the `btd_inheritance` entry to 0.

   The default is 1, which means billing-time discounts are inherited.

4. Stop and restart the CM.

About Sharing Charges and Discounts in Service Groups

Charge and discount sharing groups are groups of services that share balances. They consist of a group owner and group members.

To set up charge or discount sharing for a service group (for example, to distribute 300 minutes to all services on a phone line), make the subscription service the sharing group owner, and make the member services the sharing group members.

The service type of the subscription service must match the service type to which the shared discount applies.

Note:

All services in a sharing group must have their own balance groups so BRM can track the balances they use and share.

See "About Charge and Discount Sharing Groups" in `BRM Managing Accounts Receivable`.

About Distributing Sharing Group Balances to Member Services

Each charge and discount sharing group member has an ordered balance group. See "About Ordered Balance Groups" in `BRM Managing Accounts Receivable`.
When a subscription service is a member in a sharing group, its member services may or may not share its ordered balance group:

- A member service that is not a member of a sharing group uses the ordered balance group of its subscription service. By default, the member service inherits member status in the sharing group from the subscription service, and the member service has access to the sharing group's balances as shown in Figure 18-2.

Figure 18-2  Member Service Using Its Subscription Service's Ordered Balance Group

- A member service that has its own ordered balance group does not use the subscription service's ordered balance group. This is true even when the member service and subscription service belong to the same sharing group as shown in Figure 18-3.

Figure 18-3  Member Service Using Its Own Ordered Balance Group

- If a member service and its subscription service belong to different sharing groups, the services have access only to the shared balances in the sharing group to which they belong (see Figure 18-4).
When the subscription service and a member service are both discount sharing group owners and have members in common, the sequence of the ordered balance group list for each discount sharing group member determines whether the member service’s discounts or the subscription service’s discounts are applied first.

If a service is a member of both a charge sharing group and a discount sharing group, the discounts are used before charges are applied.

You set up discounts and define how charges are shared in PDC or Pricing Center.

To create charge and discount sharing groups, you implement BRM opcodes in your customer relationship management (CRM) system.

### About Changing Status in Service Groups

The status of the subscription service or any member service can be active, inactive, or closed. Changing the status of the subscription service changes the status of all the group’s member services.

- When you inactivate a subscription service, all its member services are inactivated.
- When you close a subscription service, all its member services are closed.
- When you reactivate a subscription service that was inactivated or closed, all its member services are reactivated.

Changing the status of a member service affects only that service. If you close a member service and then close the subscription service, reactivating the subscription service does not reactivate the independently closed member service.

Closing a service cancels all charge and discount offers owned by that service. If you reactivate a service and want to restore canceled offers, you must repurchase the offers for the service.

### About Setting Up Corporate Accounts That Use Service Groups

When a corporate account uses service groups, rating performance is affected if the company has many employees whose services share account balances.

For corporate accounts, it is better to set up a bill unit hierarchy in which the company account owns the paying parent bill unit and the employees own accounts that include the following:
• Their own subscription and member services
• Nonpaying child bill units of the corporate account’s paying parent bill unit
That way, the charges are rolled up to the corporate account at billing time, so rating performance is less affected.

About Creating and Managing Service Groups

Creating and managing service groups includes the following tasks:

• About Creating a Service Group
• About Adding Bundles to Service Groups
• About Setting Up Balance Groups for Service Groups
• About Sharing ERAs in Service Groups
• About Modifying Service Groups

About Creating a Service Group

A subscription service is a `service` object in the BRM database. You can create a custom storable class to represent a subscription service, or you can use an existing storable class. Member services can be a subclass of the subscription service or a different service type.

A service instance can be either a subscription service or a member service, but not both. After a service is selected as a subscription service, it cannot also be used as a member service in the same service group.

You can create service groups in the following ways:

• When creating a package in PDC or Pricing Center. When customers purchase the package, the member service objects are created and associated with the subscription service object. Additional member services can be added during customer account creation.
• If you implement your own CRM system, customer service representatives (CSRs) can dynamically set up service groups or add member services to an existing service group when creating customers. To create service groups when creating accounts, you implement BRM opcodes in your CRM system. See BRM Opcode Guide for more information.

About Adding Bundles to Service Groups

You can add bundles to the subscription service and to member services. A bundle owned by the subscription service is a subscription bundle. That is, the offers in the bundle are available for rating all services in the service group.

You associate bundles with subscription services when you add bundles to packages in PDC or Pricing Center.

A CSR can also add a bundle to a subscription service or a member service that a customer has purchased.
About Setting Up Balance Groups for Service Groups

Although a subscription service can use the account's balance group, tracking usage for a service group when the subscription service has its own balance group is much easier.

Note:
You cannot transfer a subscription service that uses the account's balance group to another account. See "About Transferring a Service Group to Another Account".

You can do the following:

- **Associate all member services with the subscription service balance group.**
  In this case, when customers use member services, the cycle and usage fees are stored in the subscription service balance group as shown in Figure 18-5:

  **Figure 18-5  Storage of Member Cycle and Usage Fees in Subscription Balance Group**

  **Subscription service:**

  ![Subscription service diagram]

  **Member services:**

  ![Member services diagram]

  When all member services use the subscription service balance group, all noncurrency balances, such as minutes and frequent flyer miles, are shared by all services in the group.

  For example, the Line A subscription service has two member services: a telephony service that includes 360 minutes of usage per month and an SMS service that includes 30 minutes per month. Both member services use the Line A subscription service balance group. At the beginning of the month, a balance of 390 minutes is stored in that balance group, and all services have access to the 390 minutes as shown in Figure 18-6:
Create a balance group for each member service in the group.

This makes it possible to set up shared discounts, set credit limits on service balances, and limit balance consumption to the specified service. Initially, the member service balance groups have the same credit limit, floor, and threshold as the subscription service balance group. You can later change them.

When member services have their own balance groups, cycle and usage fees for each service are tracked in the member service's balance group as shown in Figure 18-7:

When a member service has its own balance group, any noncurrency balances included with the service can be consumed by that service only.

In Figure 18-8, the SMS service includes 30 minutes that can be applied only to the SMS service, which has access only to its own balances; it cannot use the minutes included with any other service.
Figure 18-8  Noncurrency Services with Multiple Balance Groups

Subscription service:

Line A

- Line A's balance group
- Free minute balance = 360

Member services:

Telephony

- 360 free minutes

SMS

- 30 free minutes

- Data service balance group
- Free minute balance = 30

- Associate groups of services with different balance groups.

A member service can also be associated with another member service's balance group to track usage for those services together. However, member services cannot use the balance group of a service that is not in the same service group.

You might want to track some types of noncurrency balances at the subscription level (for example, the number of months users have been subscribers regardless of whether they have changed the services on their lines).

To track noncurrency balances at the subscription level, configure a counter sub-balance for the subscription service. See "About Noncurrency Sub-Balances" in BRM Creating Product Offerings.

For more information about balance groups, see BRM Configuring and Running Billing.

About Sharing ERAs in Service Groups

Any member service that does not have its own extended rating attribute (ERA) uses the subscription service's ERA. If the subscription service and member service both have an ERA, the member service's ERA is used to rate the member service's usage.

For general information about ERAs, see BRM Telco Integration.

About Modifying Service Groups

You can perform the following actions to modify a service group:

- Add a member service to the group.
- Add bundles to a service in the group.
- Change the status of a service in the group.

You can modify a service group's services when setting up services in PDC or Pricing Center.

To modify a service group's services after customers have purchased them, implement BRM opcodes in your CRM system. See BRM Opcode Guide.
About Canceling a Subscription Service

A subscription service can be canceled at any time during the billing cycle. You can also bill the account for the service immediately upon cancellation.

In BRM, a subscription service is canceled when the following occur:

- The account that owns the subscription service is closed.
- The subscription service expires.
- The subscription service is canceled by a CSR.

To perform a subscription service cancellation, BRM does the following:

- Closes the subscription service and all its member services. See "About Service Status When a Subscription Service Is Canceled".
- Cancels the charge and discount offers associated with the subscription service and its member services. See "Effect of Canceling a Subscription Service on Offers Owned by the Service".
- Deletes the sharing groups owned by the subscription service and its member services, and removes the services from any sharing groups of which they are members. See "Effect of Canceling a Subscription Service on Sharing Groups Owned by the Service".
- Bills the account if specified. See "About Billing on Subscription Service Cancellation".

CSRs can use Billing Care or Customer Center to cancel subscription services as they cancel other services. To cancel subscription services with your client application, implement BRM opcodes. See "Canceling a Service Group" in BRM Opcode Guide.

About Service Status When a Subscription Service Is Canceled

When a subscription service is canceled, all its member services are also canceled. The status of the subscription service and its member services is set to closed.

- The status-flag value of the subscription service is set to PIN_STATUS_FLAG_CANCEL_LINE, and the status-flag value of member services is set to PIN_STATUS_FLAG_DUE_TO_SUBSCRIPTION_SERVICE, which signifies that the member service status was changed due to the subscription service status change.
- The purchase, usage, and cycle end dates for the subscription service and all its member services are set to the cancellation date.

See "About Changing Status in Service Groups".

Effect of Canceling a Subscription Service on Offers Owned by the Service

Canceling a subscription service cancels all the charge and discount offers owned by the service and its member services.

- The status of the offer instances owned by the subscription service and its member services is set to canceled.
- The purchase, usage, and cycle end dates for the canceled offer instances are set to the subscription service cancellation date.
Effect of Canceling a Subscription Service on Sharing Groups Owned by the Service

A subscription service and its member services can own a sharing group, be members of a sharing group, or both.

When a subscription service is canceled, BRM does the following:

- For each sharing group member, the sharing group's balance group is deleted from the group member’s ordered balance group list.
- The subscription service and its member services are removed from any sharing groups in which they are members.

For more information, see "About Charge and Discount Sharing Groups" in BRM Managing Accounts Receivable.

About Transferring a Service Group to Another Account

You can transfer a service group from one account to another at any time during the billing cycle.

Transferring a service group transfers the subscription service and all member services. The member service objects are updated to reference the target account object.

You can transfer service groups from both closed and active accounts:

- **From a closed account to an active account**: The subscription service remains closed after the transfer. You must manually activate the service.
- **From an active account to a closed account**: The subscription service remains active after the transfer. You must manually activate the account.

All pending scheduled actions (Ischedule objects) associated with the subscription service and the member services are also transferred to the target account. Completed actions (Ischedule objects with Done status) are not transferred.

To prevent the transfer of a service group to a closed account, use event notification and write a custom opcode to generate an error when the service group transfer event occurs. See "Using Event Notification" in BRM Developer's Guide.

---

**Note:**

When you transfer a service group to another account, its balance groups are also transferred. If the subscription service or one of its member services uses the account default balance group, however, that balance group cannot be transferred. Therefore, the service group cannot be transferred. To avoid this, the subscription service and member services should use service balance groups. You can define a balance group for the subscription service and associate all the member services with that balance group. See "About Setting Up Balance Groups for Service Groups".

You cannot backdate a service group transfer.
Each time a subscription service is transferred, BRM stores information about the source account and the date and time when the transfer occurred in a transfer list. The transfer list is stored in the /service object. BRM uses information in the transfer list to rate and record delayed events and call detail records (CDRs) and to bill the source account for usage that occurred before the transfer.

To transfer service groups, you implement BRM opcodes in your client application. See BRM Opcode Guide for more information.

About Transferring Service Groups between Schemas

To transfer a service group between accounts stored in different schemas, do one of the following:

- Migrate the accounts to the same schema and then transfer the service group.
  You can use Account Migration Manager (AMM) to migrate accounts between database schemas.

  Note:
  AMM is an optional feature. Contact your BRM account manager for more information about using AMM.

- If you do not want to migrate the accounts to the same schema, enable the RecreateDuringSubscriptionTransfer business parameter. See "Transferring Subscription Services between Accounts in Multiple Schemas" in BRM Opcode Guide.

About Transferring Service Groups Associated with Objects

When a service group is transferred to another account, objects associated with the subscription service and its member services are also transferred to that account.

A service can be associated with the following objects:

- Service profiles. See "About Transferring Service Groups Associated with Service Profiles".
- Devices and phone numbers. See "About Transferring Service Groups Associated with Devices".
- Charge and discount offers. See "About Transferring Service Groups Associated with Offers".
- Charge and discount sharing groups. See "About Transferring Service Groups Associated with Sharing Groups".
- Service sub-balances and account bill units. See "About Transferring Balance Groups during Service Group Transfer".

About Transferring Service Groups Associated with Service Profiles

Service profiles store additional information about a service. A service profile is associated with a service and the account the service belongs to. When a service group is transferred to another account, profile objects associated with the subscription service and its member services are updated to reference the target account.
About Transferring Service Groups Associated with Devices

Devices such as a mobile phone are associated with services. A device object is associated with a service and the account that the service belongs to. When a service group is transferred to another account, the devices associated with the subscription service and its member services are updated to reference that account.

See "Managing Devices with BRM" in BRM Developer's Guide.

About Transferring Service Groups Associated with Offers

BRM rates events by using the charge and discount information associated with the service for which the events are generated. The purchase, cycle, and usage start and end dates define the validity periods during which a charge or discount offer can be purchased and cycle fees and usage charges can be applied.

When a service group is transferred, BRM cancels the charge and discount offers associated with the subscription service and its member services for the source account by setting the end dates to the transfer date. When delayed billing is enabled, delayed events and CDRs are rated and discounted by using the canceled charge and discount offers for the source account.

**Note:**

To rate and discount delayed events for the source account, the canceled charge and discount offers must persist in the account.

Another instance of the charge and discount offers associated with the subscription service and its member services is added to the target account with the purchase, cycle, and usage start dates set to the transfer date. The end dates are set to the original end dates. Events generated by the subscription service after the transfer date are then rated and discounted for the target account.

**Note:**

- To discount any charges associated with the charge offers, such as the prorated cycle fee amount, the discount offer instances are added to the target account before the charge offer instances.
- When a service group is transferred, the source and target accounts are not charged any purchase or cancellation fees that are generally applied when a charge or discount offer is purchased or canceled.

See "Managing Purchased Charge Offers".

About Transferring Service Groups Associated with Sharing Groups

Charge and discount sharing groups associated with a subscription service or its member services are not transferred when the service group is transferred to another account. Instead:
• **If a subscription or member service owns a sharing group**: The group is deleted, and the group’s balance group is deleted from the ordered balance group list for each member in the group. You must re-create the groups after the service is transferred to the target account.

• **If a subscription or member service is a member of a sharing group**: The service is deleted from the group, and the group’s balance group is deleted from the ordered balance group list for the service. The service must be added back to the group after the service is transferred.

See “About Sponsor Groups” in *BRM Developer’s Guide*.

### About Transferring Balance Groups during Service Group Transfer

A subscription service can have its own balance group. Each member service can also have its own balance group. The balance group is associated with the account’s bill units (*billinfo* objects).

When a service group is transferred to another account, the balance groups associated with the subscription service and its member services are also transferred to the target account. Either the balance groups are added to an existing bill unit in the target account, or a new bill unit is created for the service only.

**Note:**

- You cannot transfer a service group whose subscription service or member services use the account default balance group. To avoid this, subscription and member services should use service balance groups. See “About Creating and Managing Service Groups”.
- Usually, a balance group cannot move to another bill unit when its current bill unit has unallocated payments or adjustments, open refunds, or unresolved disputes. This restriction does not apply, however, when a balance group is moved to another account as the result of a subscription service transfer.

### About Assigning Events to Bill Items after a Service Group Transfer

Each event generated by a subscription service is assigned to a bill item (*item*). During billing, charges from all the bill items are collected in a bill for the account that owns the service group.

When a subscription service is transferred, BRM creates new bill items for the target account by using item configurations (*config/items*). Events generated after the transfer date are assigned to the new bill items. The new bill items are linked to the bill unit referenced by the service’s balance group in the target account. As a result, charges for the new events are applied to the target account.

Delayed events and CDRs are assigned to the bill items in the source account. The bill items are associated with the source account bill unit; therefore, charges for the bill items are applied to the source account.
About Transferring Noncurrency Assets during Service Group Transfer

Before a service group is transferred to another account, a noncurrency asset is stored in a single sub-balance. After the service group is transferred, each asset is prorated and stored in two sub-balances with different validity periods:

- **Original sub-balance**: By default, assets in this sub-balance are valid from the start of the source account's current billing cycle to the transfer date.
- **New sub-balance**: By default, assets in this sub-balance are valid from the transfer date to the end of the target account's current billing cycle.

**Note:**
To create separate sub-balances with different validity periods, the balance impact that grants the free assets must have a nonzero relative cycle start date. If the relative cycle offset in the cycle forward or cycle arrears charge is set to 0, a new sub-balance is not created. Instead, the assets are added to the original sub-balance and are valid forever.

Rating and discounting impact the assets in the original sub-balance for delayed events generated by the service before the transfer date. Assets in the new sub-balance are consumed for new events generated by the service group after the transfer date.

By default, the original sub-balance always expires on the transfer date. Because events generated by the target account occur after the transfer date, the target account cannot consume free assets from the original sub-balance. You can extend the validity period of the original sub-balance so that its free assets can be consumed by the target account. See "About Extending Sub-Balance Validity When a Service Group Is Transferred".

About Extending Sub-Balance Validity When a Service Group Is Transferred

When a service group is transferred to another account, noncurrency assets are prorated and stored in sub-balances with different validity periods. See "About Transferring Noncurrency Assets during Service Group Transfer".

To extend the validity period of the original sub-balance, set the `sub_bal_validity` business configuration parameter to one of the following options:

- **Cut**: This is the default. This option sets the validity period from the start of the source account's current billing cycle to the service transfer date.
- **Maintain**: This option retains the original end date of the bucket, even for charge offer cancellations or line transfers.
- **Align**: This option sets the validity period from the start of the source account's current billing cycle to the end of the target account's current billing cycle.
Note:

- Setting the `sub_bal_validity` business configuration parameter extends the validity period of the original sub-balance only. The assets in the new sub-balance are always valid from the service transfer date to the end of the target account's billing cycle.
- Extending the validity period of the original sub-balance extends only the time the free assets in the bucket are accessible and does not change the asset proration. The assets are prorated based on the service transfer date. See “About Prorating Cycle Fees When a Service Group Is Transferred”.
- If you do not configure the `sub_bal_validity` parameter, BRM sets the original sub-balance to expire on the service transfer date, and the target account cannot access this sub-balance.

To configure sub-balance validity, see “Configuring Sub-Balance Validity for Service Group Transfer”.

About Consuming Noncurrency Balances When a Service Group Is Transferred

When a service group is transferred to another account, noncurrency balances are prorated and stored in sub-balances with different validity periods. See "About Transferring Noncurrency Assets during Service Group Transfer".

The source account always consumes free balances from the original sub-balance because the events for this account occur before the service transfer date. The target account can consume balances from both the original sub-balance and the new sub-balance when the validity period for the original sub-balance is extended beyond the service transfer date. See "About Extending Sub-Balance Validity When a Service Group Is Transferred”.

The following examples demonstrate how noncurrency balances are consumed after a service group is transferred.

On January 1, Account A has a rollover balance of 50 minutes and a new grant of 200 minutes. Service Group A is later transferred from Account A to Account B on January 15. Figure 18-9 shows a timeline for the transfer:

**Figure 18-9   Example Timeline**

<table>
<thead>
<tr>
<th>Account B’s current billing cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account A’s current billing cycle</td>
</tr>
<tr>
<td>1/1</td>
</tr>
<tr>
<td>Rollover (50 free minutes)</td>
</tr>
<tr>
<td>Original sub-balance (200 free minutes)</td>
</tr>
</tbody>
</table>
Example 1

When `sub_bal_validity` is set to `Cut`, the rollover and original sub-balance buckets’ `valid_to` dates are set to the transfer date. The assets in the original sub-balance are prorated, and a new sub-balance is created with 100 minutes.

A delayed event for Account A is recorded on January 17 for 200 minutes. When the event is rated, 50 minutes are consumed from the rollover sub-balance, and 100 minutes are consumed from the original sub-balance. The balance in these two sub-balances becomes 0. Because the event occurred before January 15, it cannot consume minutes from the new sub-balances, and Account A is charged for the remaining 50 minutes of usage. Figure 18-10 illustrates this example.

Figure 18-10  Example 1

![Figure 18-10](image)

Example 2

When `sub_bal_validity` is set to `Maintain`, the rollover and original sub-balance buckets’ `valid_to` dates are extended to the end of Account A’s billing cycle.

A delayed event for Account A is recorded on January 17 (after the service transfer date) for 200 minutes. When the event is rated, 50 minutes are consumed from the rollover sub-balance, and 100 minutes are consumed from the original sub-balance. The balance in these two sub-balances becomes 0. Because the event occurred before January 15, it cannot consume minutes from the new sub-balance, and Account A is charged for the remaining 50 minutes of usage. Figure 18-11 illustrates this example.
When `sub_bal_validity` is set to `Align`, the rollover and original sub-balance buckets’ `valid_to` dates are extended to the end of Account B’s billing cycle.

An event for Account B is recorded on February 3 for 200 minutes. Because the validity period of the rollover and original sub-balance is extended until February 10, 50 minutes are consumed from the rollover sub-balance, and 100 minutes are consumed from the original sub-balance. The remaining 50 minutes are consumed from the new sub-balance. The balance in the rollover and original sub-balance buckets becomes 0, and the balance in the new sub-balance becomes 50. Figure 18-12 illustrates this example.
About Rolling Over Noncurrency Balances When a Service Group Is Transferred

You can configure rollovers so that noncurrency balances associated with cycle forward fees are rolled over for use in future cycles. See “About Rollovers” in *BRM Setting Up Pricing and Rating*.

When a service group is transferred, the free assets in the rollover and original sub-balance buckets are rolled over when `sub_bal_validity` is set to **Maintain** or **Align**.

**Note:**

The rollover and original sub-balance buckets are not rolled over when `sub_bal_validity` is set to **Cut**.

The following example demonstrates how unused balance assets are rolled over when a service group is transferred to another account. In this example:

- The service group’s subscription service grants 200 minutes valid from January 1.
- The service group is transferred to the target account on January 15.
- The billing cycle for the source account starts on the first of each month.
- The billing cycle for the target account starts on the 15th of each month.
- The `sub_bal_validity` business configuration parameter is set to **Maintain**.
- Rollover minutes from the previous cycle are 50.

The rollover rules are as follows:

- A maximum of 50 minutes can be rolled over to the next cycle.
- The maximum number of rollover cycles is 1.
- The maximum number of minutes that can be rolled over from previous months is 100.

On January 1, the source account is granted 200 minutes for the month of January. On January 15, the service group is transferred to the target account. The 200 minutes are prorated and put in two buckets with different validity dates: the original bucket (Sub-balance 1) and a new bucket (Sub-balance 2). Figure 18-13 illustrates this example.
When `pin_bill_day` is run on February 1, 50 minutes from Sub-balance 1 are rolled over. Because Sub-balance 1 is valid for a partial cycle (from January 15 to February 1 instead of January 15 to February 15), the assets are rolled over for one entire cycle. As a result, the Rollover 2 bucket is valid until March 15, which is the next accounting cycle end date for Account B.

When `pin_bill_day` is run on February 15, 50 minutes from Sub-balance 2 are rolled over into a new bucket (Rollover 3) and are valid until March 15. The target account is granted an additional 200 minutes for the new cycle (Sub-balance 3).

Figure 18-14 illustrates the billing timeline associated with this example.

Because the Rollover 2 and Rollover 3 buckets are valid after the transfer date, only the target account can access the assets in these buckets. The source account cannot consume the minutes from these buckets.

About Prorating Cycle Fees When a Service Group Is Transferred

Typically, cycle fees are applied at the beginning of a billing cycle for services provided during that cycle and are prorated when a service is purchased or canceled during a billing cycle.
However, a service group transfer is not the same as canceling the charge offer in the source account and purchasing it for the target account. Instead, a service transfer moves the service from one account to another. When a service group is transferred, the accounts are not charged with the charge offer purchase or cancellation fees, and therefore the purchase and cancellation proration generally applied when a charge or discount offer is purchased or canceled are also not applied.

Consequently, when a service group is transferred during a billing cycle, cycle fee proration is enforced based on the transfer date to charge the source account for the fees before the transfer and the target account for the fees after the transfer. The prorated cycle fees from the start of the billing cycle to the transfer date are applied to the source account, and the prorated cycle fees from the transfer date to the end of the billing cycle are applied to the target account.

See "About Proration" in *BRM Configuring and Running Billing*.

### Applying Billing-Time Discounts and Folds When a Service Group Is Transferred

Billing-time discounts and folds are applied at the end of the accounting cycle and are based on balances used during the cycle. In BRM, balances are stored and tracked in account default or service balance groups.

When a service group is moved to another account, the service balance group is also transferred to that account. As a result, billing-time folds and discounts are applied to the target account. See "About Transferring Balance Groups during Service Group Transfer".

For more information, see "About Billing-Time Discounts" in *BRM Creating Product Offerings*.

### About Rating Delayed Events When a Service Group Is Transferred

Delayed events generated by a service group that occur before the service group transfer are billed to the source account. See "About Assigning Events to Bill Items after a Service Group Transfer".

BRM rates events by using the charge and discount offers associated with the service for which the events are generated. When a service group is transferred, BRM stores the original account's charge and discount offers to rate delayed events generated by that account. See "About Transferring Service Groups Associated with Offers".

When delayed events are rated for the source account, the source account can be charged for usage when all the free assets have been consumed by the target account. This can occur when the validity period of the original sub-balance is extended beyond the service group transfer date. See "About Consuming Noncurrency Balances When a Service Group Is Transferred".

---

**Note:**

To rate delayed events accurately, you must rerate events for both the source and target account in the correct order. See "About Rerating Events" in *BRM Setting Up Pricing and Rating*. 

---

Chapter 18

**About Creating and Managing Service Groups**

18-22
How a Service Group Transfer Affects Account Migration

When an account selected for rerating is associated with a service group transfer, rerating requires both the source account and the target account to reside in the same database schema. For this reason, AMM does not allow the source account or the target account to be migrated to another schema.

For example:

- Account A is stored in Schema A.
- Account B is stored in Schema B.
- Account C is stored in Schema C.

To transfer a service group from Account A to Account B, you must move Account B to Schema A or Account A to Schema B and then transfer the service group to Account B.

After the service transfer, AMM does not allow you to migrate Account A or Account B to Schema C because rerating requires the source account and the target account to reside in the same schema. To transfer the service to Account C, you must move Account C to Schema A and then transfer the service.

Configuring Sub-Balance Validity for Service Group Transfer

To extend the validity period of the original sub-balance when a service group is transferred, run the `pin_bus_params` utility to change the `SubBalValidity` business parameter. This parameter takes one of the following values:

- Cut
- Maintain
- Align

`SubBalValidity` is a systemwide parameter. By default, this parameter is set to `Cut`. Changing the value of this parameter affects the sub-balance validity period for any service group transfer that occurs after the change. See "About Extending Sub-Balance Validity When a Service Group Is Transferred".

© Note:

If you change the `SubBalValidity` parameter value, BRM does not keep a record of the original setting. BRM uses the new setting for any subscription service transfers that occur after the change.

© Note:

If you do not configure the `SubBalValidity` parameter, BRM follows the default implementation by setting the original sub-balance to expire on the service transfer date.
To extend the validity of the original sub-balance:

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. Change `Cut` to one of the following:
   
   - **Maintain** to extend the validity of the original sub-balance to the end of the original account's current billing cycle.
   - **Align** to extend the validity of the original sub-balance to the end of the target account's current billing cycle.
   
   ```
   <SubBalValidity>Cut</SubBalValidity>
   ```
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*.

### About Billing for Service Group Usage

To bill for service group usage, all balance groups in the service group must be associated with a bill unit. A bill is produced for each bill unit.

**Note:**

If a member service's balance group is not associated with a bill unit, charges for that service are not billed.

You can create a bill unit for a service group or use the account's bill unit. If you use the account's bill unit, charges for any service owned by the account that are not part of the service group are also included in the bill.

You use Billing Care or Customer Center to create bill units and associate them with balance groups.

You can also create and manage bill units by using the BRM API. See "Managing Bill Units with Your Custom Application" in *BRM Configuring and Running Billing*.

### About Creating Multiple Bills for a Service Group

When a member service has its own balance group, you can create a separate bill for that service by associating it with a new bill unit. You can also associate several member services with a single bill unit if each member service has a balance group.
About Billing for Multiple Service Groups

When a customer owns multiple service groups, you can bill for each group separately or include charges for all groups in the same bill. To create one bill for all groups, associate the balance groups in all the groups with the same bill unit.

About Billing on Subscription Service Cancellation

Normally, you bill an account on its regularly scheduled billing day at the end of the billing cycle. You can also bill the account immediately upon a subscription service cancellation.

To bill the account immediately, specify the Bill Now option when canceling the subscription service. See “About Bill Now” in **BRM Configuring and Running Billing**.

Note:

If the Bill Now option is not specified, the account is billed for the subscription service during the regularly scheduled billing time.

When the Bill Now option is specified, the account is billed for all current charges generated by the subscription service until the cancellation date.

Delayed events for the service recorded after the cancellation date are billed on the next regularly scheduled billing date.

When the Bill Now option is specified, BRM bills the bill units associated with the subscription service. Typically, all the member services are associated with the subscription service's bill unit. However, a member service can have its own bill unit.

By default, any other service belonging to the same bill unit as the canceled subscription service or its member service's bill unit is also billed.

For more information, see “Configuring Bill Now” in **BRM Configuring and Running Billing**.

About Billing an Account When a Service Group Is Canceled during the Delayed Billing Period

When a service group is canceled during the billing delay period with the Bill Now option specified, the account is billed immediately.

Delayed events recorded after the cancellation date are billed on the regularly scheduled billing date.

In **Figure 18-15**, the subscription service is canceled during the billing delay period on February 5. Bill B1′ includes all current charges from January 1 to February 5 for both billing cycle B1 and billing cycle B2. Delayed events for billing cycle B1 recorded after February 5 are billed on February 10. Events generated for billing cycle B2 after February 5 are billed on March 10.
About Billing a Service Group after Transferring It to Another Account

A service group can be transferred at any time during the billing cycle.

At the time of billing, any cycle fees associated with the service group are prorated. The source account is billed for the prorated amount until the transfer date. The target account is billed for the remainder of the cycle. See "About Prorating Cycle Fees When a Service Group Is Transferred".

The source account is billed for service usage charges incurred during the time the service was owned by the source account, including delayed events that occurred before the transfer of the service. Events generated after the transfer are billed to the target account.

Any remaining noncurrency grants, such as minutes, are transferred to the target account. See "About Transferring Noncurrency Assets during Service Group Transfer".
Creating and Managing Profile Sharing Groups

A profile sharing group enables an account or an account’s service to share a profile with other accounts or services. A profile stores ERAs or other types of information about an account.

This document describes how to create and use profile sharing groups in your Oracle Communications Billing and Revenue Management (BRM) system.

To use this documentation, you should be familiar with extended rating attributes (ERAs). See "About Extended Rating Attributes" in BRM Configuring and Running Billing.

See also:
- Creating and Managing Business Profiles
- List of Customer Management Features
- BRM Concepts

About Profile Sharing Groups

A profile sharing group enables an account or an account’s service to share a profile with other accounts or services. A profile stores ERAs or other types of information about an account.

The group owner can be an account or a service. If an account is the owner, profiles from all the services owned by the account are available for sharing. If a specific service is the owner, only profiles of that service are available for sharing.

Note:

Only service profiles can be shared. Account profiles cannot be shared, even if the profile sharing group is owned by an account.

Profile sharing can be used to share an ERA, such as a friends and family list, to make it available to multiple accounts or services. For example, a friends and family list can be set up for a GSM service owned by one account and then shared so that a GSM service belonging to other accounts can use the same list.

Profile sharing groups can also be used to share custom profiles you create.

You can set up profile sharing groups in one of these ways:
- Using Billing Care.
- Using Customer Center.
• Using a customized third-party client application.

For guidelines that apply to both creation methods, see "Creating Profile Sharing Groups".

**Note:**

Profile sharing groups work similarly to charge and discount sharing groups, but profile sharing groups do not share balances and are not prioritized.

**About Profile Sharing Group Membership**

Only service profiles can be shared, but a profile sharing group's members can be accounts or services, as follows:

- **Account**: If an account is a member, any eligible event generated by the account can use a shared profile.
- **Service type**: When you specify a service type (for example, GSM) as a member, the events generated by all service instances within that service type are considered for shared profiles.

You can also specify that a member account that has not yet purchased a service of that type is automatically eligible for participation in profile sharing if it buys the service in the future.

**Note:**

The subtypes of the service type you specify do not become members. For example, if you specify the GSM service type as a member, specific GSM service instances such as GSM fax, GSM telephony, and so forth, do not.

- **Service instance**: When you specify a service instance (for example, a specific phone) as a member, only the events generated by that instance are eligible for the shared profiles. Specify membership using a specific service instance if you want to exclude other services of that type from profile sharing.

For example, if an account includes both work phone service and personal phone service, a friends and family ERA might apply only to the personal phone.

Use the PCM_OP_SUBSCRIPTION_POL_AUTO_SUBSCRIBE_SERVICE opcode to customize adding a service to a sharing group. See *BRM Opcode Guide*.

When setting up profile sharing groups:

- Profile sharing group members must be equivalent to the owner. If the owner is a service, then the members must be the same service type. If the owner is an account, then the members must also be accounts.
- Members do not need to have the same charge or discount offers as the owners. But to make use of a shared profile, members need a charge or discount offer.
configured to use an ERA type or ERA label in the shared profile to give special charge pricing or discounts.

• When the status of a profile sharing group's owner account is changed to inactive or closed, the shared profiles are no longer available to group members. During rating, the shared profiles are not retrieved. If an inactive owner account is activated, the shared profiles are once again available during rating.

• By default, BRM does not validate profile sharing group members. You can customize the PCM_OP_SUBSCRIPTION_POL_PREP_MEMBERS policy opcode to add validation for profile sharing groups. See BRM Opcode Guide.

Creating Profile Sharing Groups

To create a profile sharing group, you specify a group owner account or service, member accounts or services, and the list of the group owner's profiles that will be shared by the members.

You create profile sharing groups as follows:

• Using Billing Care.
• Using Customer Center.
• Using a third-party client application, see BRM Opcode Guide.

The following guidelines apply when creating a profile sharing group:

• The owner cannot be a member of its own group.
• The owner of a profile sharing group cannot also belong to a member-owned group. For example, if Account1 and Account2 both own profile sharing groups, and Account2 is a member of Account1's group, then Account1 cannot belong to Account2's group.

• If you add a service type as a member rather than a specific service instance, all instances of that service type become members. However, the subtypes of the service type do not become members.

For example, if you specify the GSM service type as a member, all instances of GSM become members, but GSM fax, GSM telephony, and so forth do not.

Adding a Profile Group to a Member's Ordered Balance Group

The /ordered_balgrp object stores the list of sharing groups to which an account or service belongs. This object is used with all types of sharing groups, but its significance varies:

• For charge and discount sharing groups, /ordered_balgrp controls the order in which the group's balances are impacted by events. See "How Discounts and Charges Are Applied" in BRM Managing Accounts Receivable.

• For profile sharing groups, the order is not significant, so they are added to the end of the list in the PIN_FLD_ORDERED_BALGROUPS array.

When a profile sharing group is created or modified, the PCM_OP_SUBSCRIPTION_POL_AUTO_SUBSCRIBE_MEMBERS policy opcode
automatically adds the group to the `ordered_balgrp` object of each account or service that is a member.

Modifying Profile Sharing Groups

You can modify profile sharing groups by using Billing Care or Customer Center, or by customizing a third-party application. You can modify a profile sharing group in the following ways:

- **Add members to the group**: When a member is added to the group, shared profiles become available to events generated by that member.

- **Add shared profiles**: When profiles are added to a profile sharing group, the profiles automatically are available to group members. The added profiles must be owned by the group owner and have valid dates.

- **Delete members from the group**: When a member is deleted from the group, the group owner's profiles are no longer considered in rating or discounting the former member's events.

- **Delete shared profiles**: A deleted shared profile is removed from the group's profiles list and is no longer available to members.

- **Change the owner of the profile sharing group**: When the owner of a profile sharing group changes, members use the new owner's shared profiles instead of the old owner's shared profiles.

Deleting Profile Sharing Groups

A profile sharing group is deleted when the group owner's account is closed or the sharing group is deleted by a CSR. When a profile sharing group is deleted, members can no longer use the profiles that had been shared by the owner.

When a profile sharing group is deleted, BRM also deletes the profile sharing group from each member's ordered balance group. For more information about ordered balance groups, see "Adding a Profile Group to a Member's Ordered Balance Group".

You delete profile sharing groups as follows:

- Using Billing Care.
- Using Customer Center.
- To delete a profile sharing group through a third-party client application, see *BRM Opcode Guide*.

To delete members or profiles from a group, see "Modifying Profile Sharing Groups".
Managing System and Account Currencies

This document describes how to set up the Oracle Communications Billing and Revenue Management (BRM) system currencies and account currencies.

This document includes these topics:

• About System and Account Currencies
• Setting the System Currency
• Setting the Default Account Currency
• Supporting a New Currency
• Using Multiple Currencies in Your Product Offerings
• Changing Currency Conversion Rates

See also:

• List of Customer Management Features
• BRM Concepts

About System and Account Currencies

You specify two types of currency in your BRM system:

• System currency
• Account currency

The system currency is the default currency for your entire database. It is typically the currency of the country where your BRM system is installed. Every BRM installation includes a system currency. The default system currency is US Dollars (ISO code 840).

The account currency is the currency for a specific customer account. For example, a customer living in France has euro as the account currency. All payments, balance impacts, and billing adjustments use the customer's account currency.

The account currency is set by the customer or customer service representative (CSR) when the account is created.
Note:

- You cannot change an account currency after the account has been created.
- An account can have only one primary currency. If a customer requires two primary currencies, you need to create two accounts for that customer.
- A nonpaying bill unit (/billinfo object) or account must use the same account currency as the parent account.

Your price list might include balance impacts in the system currency, which might be different from a customer's account currency. If so, BRM needs to convert the system currency that is used when rating to the customer's account currency before making an impact on the customer's account balance. Currency conversion takes effect in real time by using the following process:

1. When an event is rated, the balance impact is calculated by using the system currency.
2. The balance impact is converted to the customer's account currency and added to the balance.

By default, when rating an event, BRM tries to use a charge created for the account currency. If BRM cannot find a charge that uses the account currency, BRM searches for one that uses the system currency. If it cannot find one that uses the system currency, it continues to the next charge offer in priority order to rate the event.

Setting the System Currency

The default system currency is US Dollars (ISO code 840). To use US Dollars as the system currency, you do not have to do anything.

Note:

If your system currency is not US Dollars, you must set the system currency during installation. After installing BRM, you cannot change the system currency unless you use an SQL query, a method that BRM does not support.

To set the system currency to a currency other than US Dollars, do not preconfigure BRM during installation. Instead, edit the $PIN_CONF_SYS_CURRENCY entry in the pin_setup.values file and run the setup script.

Example of changing the system currency to the euro:

$PIN_CONF_SYS_CURRENCY = 978;

For more information, see "Installing BRM" in BRM Installation Guide.
Setting the Default Account Currency

The default account currency sets the account currency for an account when the currency is not specified at account creation.

1. Open the Connection Manager (CM) configuration file (BRM_home/sys/cm/pin.conf). BRM_home is the directory in which the BRM server software is installed.

2. Change the value of the currency entry.
   
   For example, to change currency from US Dollars to British Pounds, change 840 to 826:
   
   - fm_cust_pol currency 826

3. Save and close the file.

4. Stop and restart the CM.

Supporting a New Currency

To add a currency balance element, use PDC or Pricing Center.

For more information, see “Configuring Balance Elements” in BRM Creating Product Offerings or Pricing Center Help.

To support a currency, you need to know its ISO currency code, for example, 840 for US Dollars. To find ISO currency codes, refer to the ISO 3166 standard.

Note:

The ISO currency codes are listed in the BRM_home/include/pin_currency.h file.

Using Multiple Currencies in Your Product Offerings

When you create your product offerings, you specify the currency for charges. Most charges use the system currency.

Customers can purchase charge offers that are rated with the system currency or with their account currency. The balance impacts, however, must use the account currency. There are two ways to accomplish this:

- Allow BRM to translate all currencies to the account currency.
- Create charge offers for each currency.

Converting currencies can result in fluctuating charges based on fluctuating exchange rates. Table 20-1 shows how exchange rates can cause variations in how much a customer pays for the same event, even if the amount charged by the charge offer stays the same.
Table 20-1 Effect of Exchange Rate Variation

<table>
<thead>
<tr>
<th>Charge in System Currency</th>
<th>Exchange Rate</th>
<th>What Customer Pays in Account Currency</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1 - 1.1</td>
<td>11</td>
</tr>
<tr>
<td>10</td>
<td>1 - 1.2</td>
<td>12</td>
</tr>
<tr>
<td>10</td>
<td>1 - .9</td>
<td>9</td>
</tr>
</tbody>
</table>

If you are contractually obligated to maintain consistent charges, you might need to create charge offers in multiple currencies.

For example, to charge for broadband access time, you could create charge offers that include broadband usage charge pricing with different currency balance impacts. There are advantages to this approach:

- You do not have to update the currency conversion rates.
- Your customers are billed in consistent amounts that do not change with exchange rates.
- When customers create an account, they see prices in a familiar currency.

Note:

To create multiple charge offers using different currencies, create a set of charge offers for one currency, and then use the Product Creation wizard to copy the charge offers for each currency.

Changing Currency Conversion Rates

BRM stores conversion rate information in the `/config/currency/conversionrates` object. You must update the object when conversion rates change.

The object can store conversion rates that apply to multiple time periods. For example, you could have different rates for the period from January 1, 2009, to July 31, 2009, and the period from August 1, 2009, to October 31, 2009.

At CM startup, BRM caches the `/config/currency/conversionrates` object. To support accurate conversion for suspended payments, the entire object, including all conversion rates, is cached. When you recycle a suspended payment, BRM uses the conversion rate that was valid for the period when the payment was originally made.

It's important therefore to configure the `/config/currency/conversionrates` object to include conversion rates for all time periods that might be relevant to suspended payments.
Note:

Include *only* conversion rates for time periods that are required to support suspended payments. If the `/config/currency/conversionrates` object includes a very large number of conversion rates, caching it can consume significant amounts of memory.

By default, the conversion rates for EMU currencies to euro and for euro to EMU currencies is preconfigured and loaded into the database when BRM is installed.

You define the conversion rates and their time periods by adding PIN_FLD_CUR_RATES_TIMEFRAMES arrays to the `/config/currency/conversionrates` object. There is a separate array for each time period. The rates themselves are defined in the PIN_FLD_CUR_CONV_RATES array in PIN_FLD_CUR_RATES_TIMEFRAMES. The time periods are defined by PIN_FLD_START_T and PIN_FLD_END_T fields.

Note:

Stop and restart the CM after editing the object.
Supporting EMU Currencies and the Euro

This document describes how to manage Euro currencies in Oracle Communications Billing and Revenue Management (BRM).

See also:

- Managing System and Account Currencies
- List of Customer Management Features
- BRM Concepts

Supporting EMU Currencies and the Euro

Countries that joined the Economic and Monetary Union (EMU) before February 2002, can only use the euro as their legal currency. Countries that joined the EMU after February 2002 can still use their national currency and the euro during the crossover period. By default, accounts in crossover countries use a primary account currency and a secondary account currency. These two account currencies handle the process of converting to the euro. They allow customers to pay in euros or in their native EMU currency until the conversion to the euro is complete.

When customers in EMU member countries create an account, they can select either the euro or the EMU as their primary account currency.

- When the primary account currency is an EMU currency, BRM automatically assigns the euro as the secondary account currency.
- When the euro is used as the primary account currency, the customer can choose an EMU currency as the secondary currency, or choose no secondary currency.

**Note:**

You can specify whether BRM requires a secondary currency when the euro is the primary currency. See Changing Currency Conversion Rates.

Using the Euro and EMU Currencies in Your Price List

You cannot convert directly from one EMU currency to another EMU currency. You must use triangulation, that is, convert from one EMU currency to the euro and then from the euro to the other EMU currency. There are several implications of using triangulation:

- If you have customers from many countries, you should use the euro as the system currency. That way, all currency conversion is between the euro and EMU currencies, and never between two EMU currencies, because you cannot convert between account currencies.
If you must use an EMU currency as the system currency, you should make sure that customers who have an account currency different from the system currency cannot purchase charge offers by using the system currency. To do so, use the euro as the primary currency for those accounts.

- Restrict all charge pricing to use either the euro currency or the national EMU currencies, but not both. This ensures you do not redefine the EMU to euro conversion ratio and charge two different amounts for the same service.

Handling Euro Conversion Rounding Errors

Converting between the euro and EMU currencies can result in rounding discrepancies. To allow BRM to accept these rounding discrepancies without reporting an error, you define currency error-tolerance values for EMU currencies.

You set the currency error tolerance separately for each EMU currency. It can be based on a percentage or on minimum and maximum amounts of the total amount billed.

Use PDC or Pricing Center to set error tolerance.

**Example of percentage tolerance**

You might set the percentage error tolerance amounts for French Francs to 98%:

- If a customer pays in euros for a 100 FF charge, and the euro amount converts to 99 FF, the payment is accepted.
- If a customer pays in euros for a 100 FF charge, and the euro amount converts to 97 FF, the payment is not accepted.

**Example of amount tolerance**

You might set the minimum and maximum error tolerance amounts for French Francs to 3:

- If a customer pays in euros for a 50 FF charge, and the euro amount converts to 49 FF, the payment is accepted.
- If a customer pays in euros for a 50 FF charge, and the euro amount converts to 46 FF, the payment is not accepted.

**Example of percentage and amount tolerance**

If you set both an error tolerance percentage and a tolerance amount, the tolerance percentage overrides the tolerance amount. For example, you might set the percentage tolerance to 1% and the minimum amount tolerance to 5. The customer pays in euros for a 1000 FF charge, and the euro amount converts to 992. The percentage tolerance is met, so the payment is accepted, even though the amount minimum has not been met.

Rounding Errors for Overpayments and Underpayments

By default, BRM ignores conversion errors for overpayments and credits the excess to the customer's account. For underpayments, BRM uses the tolerance settings if the customer pays in the secondary currency. You can change how BRM uses the tolerance settings in PDC or Pricing Center.
In addition, you can specify how tolerance values are applied to primary and secondary currencies. See the `underdue_tolerance` and `overdue_tolerance` entries in the CM configuration file in `BRM_home/sys/cm`.

Changing Supported Secondary Account Currencies

To limit the EMU currency choices, edit the `config/currency/supportedcombinations` object.

Make the following changes to the object:

- Delete entries for currencies that you do not want displayed.
- Delete the `PIN_FLD_CREATED_T` field.
- Delete the `PIN_FLD_MOD_T` field.
- (Optional) To use the euro as the primary account currency without using a secondary currency, change the secondary currency required field to 0:

  ```
  2   PIN_FLD_CUR_SECONDARY_REQ   ENUM [0] 1
  ```

**Note:**

To set the default secondary currency, move the currency entry to the top of the list of supported currencies. If a secondary currency is required, and the secondary account currency is not supplied during account creation, the default secondary currency is the first supported currency listed.

**Note:**

To verify that you changed the fields, read the object by using the `testnap` utility or by displaying the `config/currency/supportedcombinations` object in the Object Browser.

**Note:**

Stop and restart the CM after editing the object.
Creating and Managing Customer Segments

A customer segment is a user-defined customer description that can be used to group accounts according to customer billing and payment practices, such as reliable bill payer.

This document describes the Oracle Communications Billing and Revenue Management (BRM) customer segment feature.

See also:
• Creating and Managing Business Profiles
• List of Customer Management Features
• BRM Concepts

About Customer Segments

A customer segment is a user-defined customer description that can be used to group accounts according to customer billing and payment practices, such as early bill payer, reliable bill payer, and delinquent bill payer. You can use customer segments to charge payment fees, provide payment incentives, and suppress bills.

Table 22-1 lists customer segments you can use to implement the following rules for early bill payers who rarely make late payments:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Payment rule</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payment incentive</td>
<td>If the payment is received 2 weeks early.</td>
<td>Apply $2 discount to the bill.</td>
</tr>
<tr>
<td>Payment fee</td>
<td>If the payment fails due to expired credit card.</td>
<td>Override the $2 payment fee.</td>
</tr>
<tr>
<td>Bill suppression</td>
<td>If the bill is less than $10.</td>
<td>Suppress the bill until the end of the next billing cycle. <strong>Note:</strong> To implement automatic bill suppression, you must first define customer segments and then associate bill suppression settings with them. See “About Bill Suppression” in BRM Configuring and Running Billing.</td>
</tr>
</tbody>
</table>

An account can belong to more than one customer segment.
Defining Customer Segments

You configure customer segments in the BRM_home/sys/data/config/pin_customer_segment.xml file, where BRM_home is the directory in which the BRM server software is installed.

To create a customer segment, add a CustomerSegment child element to the CustomerSegments element. Each CustomerSegment child element is identified by an ID and a character string. For example:

```xml
<CustomerSegmentConfiguration>
  <CustomerSegments>
    <CustomerSegment ID="1001">Early bill payer</CustomerSegment>
    <CustomerSegment ID="1002">Reliable bill payer</CustomerSegment>
    <CustomerSegment ID="1003">Late bill payer</CustomerSegment>
  </CustomerSegments>
</CustomerSegmentConfiguration>
```

The ID and string items are described in Table 22-2:

<table>
<thead>
<tr>
<th>XML items</th>
<th>Description</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>A number that identifies the customer segment in the BRM database. An account belongs to a customer segment when that segment's ID is added to the /account object's PIN_FLD_CUSTOMER_SEGMENT_LIST field.</td>
<td>Specify any integer greater than 1000.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If a customer segment is not defined for an account, the default value of 0 is used. All accounts belong to this customer segment.</td>
</tr>
<tr>
<td>string</td>
<td>A character string that describes the type of accounts in the customer segment (for example, reliable bill payer or delinquent bill payer).</td>
<td>Minimum length is 0 characters. Maximum length is 1023 characters.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> This string is mapped to the PIN_FLD_DESCR field in the /config/customer_segment object.</td>
</tr>
</tbody>
</table>

Run the "load_pin_customer_segment" utility to load the contents of the pin_customer_segment.xml file into the /config/customer_segment object in the BRM database.

Note:

The load_pin_customer_segment utility needs a configuration file (pin.conf) in the directory from which you run the utility. See "Creating Configuration Files for BRM Utilities" in BRM System Administrator's Guide.
To load customer segments into BRM:

1. Open the `pin_customer_segment.xml` file (`BRM_home/sys/data/config`) by using an XML editor or a text editor.
2. Edit the file.
3. Save the file.

To validate the XML schema in your file, run the "load_pin_customer_segment" utility with the `-t` parameter

This parameter validates the contents against the XML file’s schema definition before loading the data into the `config/customer_segment` object.
Implementing Customer Segment Information

To implement customer segments, create your own application or use a third-party client application to assign customer segments during account maintenance.

After you define customer segments in BRM, any rules you define for a segment apply to all accounts that belong to that customer segment. For example, if you define payment fees or payment incentives based on a customer segment, all accounts that belong to that segment, and conform to any additional criteria, will receive the fee or incentive.

To add customer segments to an account during account maintenance, include them in the PCM_OP_CUST_UPDATE_CUSTOMER input flist. The segments are in the PIN_FLD_CUSTOMER_SEGMENT_LIST field in the PIN_FLD_ACCTINFO array.

Note:
Accounts created by a custom application are assigned a customer segment value of 0.
Creating and Managing Business Profiles

Use business profiles to classify types of data to specify how to treat the data. For example, business profiles can be used to classify bills as prepaid or postpaid to determine how bills and payments should be handled.

This document explains how to manage Oracle Communications Billing and Revenue Management (BRM) business profiles.

See also:
- Creating and Managing Customer Segments
- List of Customer Management Features
- BRM Concepts

About Business Profiles

Use business profiles to classify types of data to specify how to treat the data. For example, business profiles can be used to classify bills as prepaid or postpaid to determine how bills and payments should be handled.

A business profile includes the following:

- A set of requirements that a bill unit and its related objects must meet to be associated with the profile. These requirements are stored in validation templates linked to the business profile. You can create validation templates for the following objects:
  - /account
  - /balance_group
  - /billinfo
  - /group/sharing/charges
  - /group/sharing/discounts
  - /group/sharing/profiles
  - /ordered_balgrp
  - /profile/acct_extrating
  - /profile/serv_extrating
  - /purchased_discount
  - /purchased_product
  - /service
- An array of key-value pairs used to retrieve information about the bill units that belong to the business profile. For example, a business profile object that includes the key Prepaid with the value Yes indicates that bill units belonging to the business profile are prepaid. A business profile set up for postpaid bill units
includes the key **Postpaid** with the value **Yes**. To find out whether a bill unit is associated with a prepaid or postpaid bill, use the PCM_OP_CUST_GET_BUSINESS_PROFILE_INFO opcode to check the value of this key in the bill unit’s business profile. See **BRM Opcode Guide** for information.

**Note:**

To use business profiles, create a custom user interface (UI) that does the following:

- Enables customer service representatives (CSRs) to assign bill units to business profiles.
- Provides different interfaces based on the business profile of a bill unit. For information about setting up business profiles and assigning bill units to them, see "Creating and Managing Business Profiles".

A bill unit can belong to only one business profile at a time. To assign a bill unit to a business profile, BRM puts the Portal object ID (POID) of the business profile object into the PIN_FLD_BUSINESS_PROFILE_OBJ field of the /billinfo object.

Before assigning a bill unit to a business profile, BRM verifies that the bill unit and all of its associated objects comply with the requirements specified in the business profile's validation templates.

To create business profiles and validation templates, you edit the `BRM_home/sys/data/config/pin_business_profile.xml` business profile configuration file, where `BRM_home` is the directory in which the BRM server software is installed. This file contains three default business profiles that can be used to set up cache residency distinction. See "Editing the Business Profile Configuration File".

After editing the configuration file, you use the "load_pin_business_profile" utility to load the file’s contents into the BRM database. See "Setting Up Business Profiles and Validation Templates".

### Setting Up Business Profiles and Validation Templates

To create, modify, or delete business profiles (`/config/business_profile` objects) and validation templates (`/config/template` subclass objects), edit the business profile configuration file (`pin_business_profile.xml`) and then load its contents into the BRM database:

1. Open the `pin_business_profile.xml` file in an XML editor or a text editor. By default, the file is in the `BRM_home/sys/data/config` directory.
2. Enter the appropriate information into the file. See "Editing the Business Profile Configuration File".
3. Save and close the file.
4. Use this command to load `pin_business_profile.xml` file:

    ```
    load_pin_business_profile pin_business_profile.xml
    ```
Note:

- When you run the utility, the `pin_business_profile.xml` and `business_configuration.xsd` files must be in the same directory. By default, both files are in `BRM_home/sys/data/config`.
- This utility needs a configuration (`pin.conf`) file in the directory from which you run the utility.
- If you do not run the utility from the directory in which `pin_business_profile.xml` is located, include the complete path to the file. For example:

  ```
  load_pin_business_profile BRM_home/sys/data/config/pin_business_profile.xml
  ```

5. To verify that the business profile and validation template information was loaded, display the `/config/business_profile` objects and `/config/template` subclass objects by using one of the following features:

  - Object Browser
  - `robj` command with the `testnap` utility

For information about reading an object and writing its contents to a file, see "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.

**Editing the Business Profile Configuration File**

You configure all of the business profiles (`/config/business_profile` objects) and validation templates (`/config/template` subclass objects) in your BRM system in the `BRM_home/sys/data/config/pin_business_profile.xml` file. This file contains prepaid, postpaid, and convergent business profiles by default.

To edit this configuration file, open it in an XML editor or text editor and then perform these tasks:

- Defining Business Profiles
- Modifying Business Profiles
- Deleting Business Profiles
- Deleting Validation Templates

**Defining Business Profiles**

In the business profile configuration file, business profiles (`/config/business_profile` objects) are defined as **BusinessProfile** child elements of the **BusinessProfileList** parent element. A **BusinessProfile** child element consists of the following elements and attributes:

- The name of the business profile ("BusinessProfile name").
- A description of the business profile ("Desc").
- A list of associated validation templates ("TemplateId" elements).
• A list of key-value pairs ("NameValue" elements). An unlimited number of key-value pairs can be associated with a business profile. See "About Business Profiles".

• An optional attribute ("type"). If the type attribute value is Invoice, the business profile is identified as an invoicing business profile. See "Specifying BI Publisher Invoice Report and Template Names in BRM" in BRM System Administrator's Guide.

The syntax of a BusinessProfile child element is:

```xml
<BusinessProfileList>
  <BusinessProfile name="string" type="string">
    <Desc>string</Desc>
    <!-- Specify List of Templates -->
    <TemplateId name="string" type="object_type" />
    <TemplateId name="string" type="object_type" />
    <TemplateId name="string" type="object_type" />
    <!-- Specify List of Key Values -->
    <NameValue key="string" value="string" />
  </BusinessProfile>
</BusinessProfileList>
```

To create a business profile, add a BusinessProfile child element to the BusinessProfileList parent element. In the child element, specify values for the items listed in Table 23-1:

**Table 23-1 Business Profile Elements**

<table>
<thead>
<tr>
<th>XML element or attribute</th>
<th>Description</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>BusinessProfile name</td>
<td>Character string used as the name of the business profile object (for example, PrepaidGold).</td>
<td>The name must be unique within your BRM system. Minimum length is 1 character. Maximum length is 255 characters. <strong>Note:</strong> This string is mapped to the PIN_FLD_NAME field in the /config/business_profile object. Values in this field can be used to populate a list of business profiles in a user interface (UI). When creating the string, take any UI length restrictions into consideration.</td>
</tr>
<tr>
<td>type</td>
<td>An optional attribute that identifies a business profile.</td>
<td>Invoice</td>
</tr>
<tr>
<td>Desc</td>
<td>Character string that describes the type of bill units that belong to the business profile (for example, Bill units with credit balances).</td>
<td>Minimum length is 1 character. Maximum length is 255 characters. <strong>Note:</strong> This string is mapped to the PIN_FLD_DESCR field in the /config/business_profile object. Values in this field can be used to populate a list of business profile descriptions in a UI. When creating the string, take any UI length restrictions into consideration.</td>
</tr>
<tr>
<td>XML element or attribute</td>
<td>Description</td>
<td>Possible values</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------</td>
<td>----------------</td>
</tr>
</tbody>
</table>
| TemplateId               | ID of a validation template associated with the business profile. In each TemplateId element, specify the following:  
• The name value of a validation template defined in the file's TemplateList element.  
• The type value of the same template. | - |
| NameValue                | Key-value pair used to retrieve information about objects associated with the business profile. In each NameValue element, specify the following:  
• key: Character string used with NameValue value to describe characteristics of the bill units belonging to the business profile.  
• value: Character string used with NameValue key to describe characteristics of the bill units belonging to the business profile (for example, Yes and No). | Minimum length of each character string is 1 character.  
Maximum length of each character string is 255 characters. |

### Modifying Business Profiles

To modify a business profile already defined in your BRM system, redefine the business profile in your business profile configuration file (see "Defining Business Profiles"). You can change any aspect of a business profile except "BusinessProfile name".

When you load the contents of the configuration file into your BRM database (see "Setting Up Business Profiles and Validation Templates"), BRM incorporates your changes into the existing business profile object.
Deleting Business Profiles

To delete a business profile from your BRM system, add the following child element to the BusinessProfileList element in your business configuration file:

```xml
<BusinessProfile name="name_of_profile_to_delete" action="delete">
```

---

**Note:**
You cannot delete a business profile to which any bill unit belongs.

When you load the contents of the configuration file into your BRM database (see "Setting Up Business Profiles and Validation Templates"), BRM deletes the specified business profile.

Deleting Validation Templates

To delete a validation template from your BRM system, add the following child element to the TemplateList element in your business configuration file:

```xml
<Template name="name_of_template_to_delete" type="type_of_template_to_delete" action="delete">
```

---

**Note:**
You cannot delete a validation template associated with a business profile.

When you load the contents of the configuration file into your BRM database (see "Setting Up Business Profiles and Validation Templates"), BRM deletes the specified validation template.
Masking Sensitive Customer Data

This document describes how to mask sensitive customer data in Oracle Communications Billing and Revenue Management (BRM).

See also:

- List of Customer Management Features
- BRM Concepts

About Masking Sensitive Customer Data

You can prevent access to and logging of sensitive customer data, such as banking information and passwords, by masking the string data fields that store this information. BRM client applications, transaction messages, and logs can contain sensitive information.

Note:

By default, data masking is enabled to protect sensitive data.

Masking Sensitive Data in Logs When Using Client Applications

Portal Communication Module (PCM) C++ and PCM Java client applications log flists containing sensitive customer data before calling the CM for processing. Client application logs can contain flists in either the standard BRM format or in XML format. Standard format flist fields configured for masking are automatically hidden before logging by the PIN_FLIST_TO_STR macro. To mask sensitive data in flists stored in XML format during logging, your client application must call the XMLToFlist class included in the pcm.jar file.

For more information on developing PCM C++ and Java client applications see "Creating Client Applications by Using PCM C++" and "Creating Client Applications by Using Java PCM" in BRM Developer's Guide.

Masking Additional Fields in Client Application Logs

BRM masks the following string data fields in client application logs for standard format flists and XML flists processed by the XMLToFlist class:

- PIN_FLD_PASSWD
- PIN_FLD_PASSWD_CLEAR
- PIN_FLD_DEBIT_NUM
You can add additional fields to be masked, including custom fields, in client application logs based on your security requirements. A list of default BRM fields are defined in the `pin_flds.h` file while custom fields are found in the `cust_flds.h` file. See "Creating Custom Fields and Storable Classes" in BRM Developer’s Guide for information on adding custom fields.

To mask additional fields in logs of PCM C++ applications:

1. Open the `BRM_home/include/pin_flds.h` file or the `BRM_home/include/cust_flds.h` file and obtain the field IDs you want to mask.

2. For each field requiring masking, add a line at the end of either the `pin_flds.h` or `cust_flds.h` file using the following syntax:

   ```
   #define Field_Name PIN_MAKE_FLD(Field_Name, ID)
   Field_Name masked
   ```

   where `Field_Name` is the string field to mask in PCM C++ client application logging and `ID` is the BRM assigned ID for the field. For example, mask the `PIN_FLD_CHECK_NO` field by using the following line:

   ```
   #define PIN_FLD_CHECK_NO PIN_MAKE_FLD(PIN_FLDT_STR, 931)
   PIN_FLD_CHECK_NO masked
   ```

   **Note:**

   Ensure that your `cust_flds.h` file does not contain duplicate entries.

3. Save the file.

4. If only masking additional fields in `pin_flds.h`, restart the CM.

To mask fields defined in `cust_flds.h`, complete the following additional steps:

1. Make a copy of your `cust_flds.h` file named `masked_fields`.

2. Run the `BRM_home/bin/parse_custom_ops_fields.pl` perl script which generates the `masked_fields.dat` file using the following syntax:

   ```
   perl -S parse_custom_ops_fields.pl -L pcmc -I masked_fields -O masked_fields.dat
   ```

3. Add the following entry in the `pin.conf` file for your PCM C++ client application:

   ```
   - - ops_fields_extension_file path/masked_fields.dat
   ```

   where `path` is the path to the `masked_fields.dat` file.
Note:

The `pin.conf` file must only have one `-- ops_fields_extension_file` entry. If you already have a `cust_flds.h` file, append your masking entries in the same file and generate a single `masked_fields.dat` file.

4. Restart the CM.

To add additional string data fields for masking in logs of PCM Java applications:

1. Open the `BRM_home/include/pin_flds.h` file or the `BRM_home/include/cust_flds.h` file and obtain the field IDs. BRM assigns each field a numerical value listed at the end of each row. Use this field ID in the `Infranet.properties` file for masking. For example, the field ID for the `PIN_FLD_CHECK_NO` field is `931`:

   ```
   #define PIN_FLD_CHECK_NO PIN_MAKE_FLD(PIN_FLDT_STR, 931)
   ```

2. Open the `Infranet.properties` file for your PCM Java application in a text editor.

3. Add a line for each additional field to be masked using the following syntax:

   ```
   infranet.custom.masked.field.field_id=masked
   ```

   where `field_id` is the field ID for the default or custom field you want to mask.

4. Save the file.

5. Verify that the `Infranet.properties` file is included in the classpath of the PCM Java client application process.
Validating Account Data

This document describes how to analyze data from the Oracle Communications Billing and Revenue Management (BRM) database.

See also:

• Collecting Information About Customers
• List of Customer Management Features
• BRM Concepts

About Validating Account Information

The Account Dump utility (ADU) is a diagnostics tool that enables you to validate account information before or after certain business processes, for example, after completion of a migration or upgrade or before billing or payment allocation.

Note:
ADU can be performance intensive, depending on the number of accounts for which data is being retrieved and the volume of data being processed. Avoid running performance-intensive operations, such as billing, while running ADU.

The account dump and data validation process is as follows:

1. Create an input file with the account search specification. ADU uses the specification to select accounts in the BRM database. See "About ADU Account Search".

2. ADU searches the accounts in the BRM database, retrieves the related object information, and dumps the information to an output file. See "About ADU Account Dump".

3. ADU validates the contents of the output file. See "About Account Data Validation".

About ADU Account Search

You can request ADU to dump information for a single account or for multiple accounts. To specify the accounts for which you want to dump information, you provide as input a text file that contains the account search specification. The search specification must be in the form of an flist.

For example, the following search flist requests the dump of the account 0.0.0.1 / account 789888 0:
The following search flist requests the dump of all the accounts with billing day of month (DOM) as 1:

0 PIN_FLD_POID      POID [0] 0.0.0.1 /search/pin -1 0
0 PIN_FLD_FLAGS     INT [0] 256
0 PIN_FLD_TEMPLATE  STR [0] "select X from /billinfo where F1 = V1 "
0 PIN_FLD_RESULTS   ARRAY [0]
1    PIN_FLD_ACCOUNT_OBJ POID [0] NULL
0 PIN_FLD_ARGS      ARRAY [1]
1    PIN_FLD_ACTG_CYCLE_DOM INT [0] 1

Note:

- ADU considers each account as a standalone. If a group owner account is specified in the search flist, ADU dumps only the contents of the owner account. ADU will not dump the contents of the group member accounts. Similarly, if a group member account is specified in the search flist, only the contents of the member account is dumped. ADU will not dump the contents of the owner account or other group member accounts.

- The PIN_FLD_RESULTS array in the search flist must contain only PIN_FLD_ACCOUNT_OBJ.

Note:

ADU can be performance intensive, depending on the number of accounts for which data is being retrieved and the volume of data being processed. Run ADU in -report mode first to determine the volume of data to be processed so that you can optimize your account search for best performance. See “pin_adu_validate” in BRM Developer’s Guide.

About ADU Account Dump

ADU provides the flexibility to choose the object information that you want dumped for an account. For example, you can request ADU to dump the /account, /service, and /invoice object information only.

You can also select the fields of an object that you want dumped. ADU then dumps only those fields into the output file. For example, you can request to dump only the first and last names of the account.

Some objects, such as /audit and /event, contain large volumes of data. Searching and retrieving information from these objects can be system intensive. Therefore, ADU provides the option to select data from these objects by specifying a date range. For example, you can configure ADU to dump the /event objects of an account updated between February 1, 2007, and March 1, 2007. For more information on using the date range option, see “Limiting Dump Information by Specifying a Date Range”.
A separate output file is generated for each account. ADU uses the format `account_POID_ID0.File_Extension` to generate the output file name where `POID_ID` is the BRM account object identifier and `File_Extension` is the extension defined in the ADU configuration file. The file extension can be configured in the ADU configuration file (`pin.conf`). See "Configuring ADU".

**Note:**
The output file is overridden if the dump is requested for the same accounts more than once and is stored in the same output folder.

By default, ADU dumps the object contents in the output file in XML format. If you prefer a different output format (for example, CSV), you can customize the output format by modifying the PCM_OP_ADU_POL_DUMP policy opcode. See .

### Limiting Dump Information by Specifying a Date Range

You can limit the dump for the following objects in the BRM database by specifying a date range:

- `/audit`
- `/bill`
- `/item`
- `/event`
- `/invoice`
- `/balance_group`

These objects normally contain large volumes of data. To limit the amount of data retrieved from these objects, use the date range option to select only data updated during a specified period.

For example, if you choose to dump the contents for `/account`, `/service`, `/bill`, and `/item` and specify February 1, 2007, as the `dump_start_time` and March 1, 2007, as the `dump_end_time`, ADU uses that date range for searching and retrieving data from the `/bill` and `/item` objects only. The date range is not used for retrieving data from the `/account` and `/service` objects.

The date range is mapped to the object date fields as follows:

- For the `/audit` object, ADU selects only those objects whose `created_t` or `effective_t` is between `dump_start_time` and `dump_end_time`.
- For the `/bill` object, ADU selects only those objects whose `end_t` is between `dump_start_time` and `dump_end_time`.
- For the `/item` object, ADU selects only those objects whose `effective_t` is between `dump_start_time` and `dump_end_time`.
- For the `/event` object, ADU selects only those objects whose `end_t` is between `dump_start_time` and `dump_end_time`.
- For the `/invoice` object, ADU selects only those objects whose `created_t` is between `dump_start_time` and `dump_end_time`. 
For the `/balance_group` object, ADU selects only those objects whose `effective_t` is between `dump_start_time` and `dump_end_time`.

To configure the date range, see "Configuring ADU".

About Account Data Validation

ADU performs two types of validations on the object contents: structural and dynamic.

**Structural validation** validates the structure of the account. For example, it validates that the POID of the parent `bill` object exists in the nonpaying child `bill` object.

**Dynamic validation** validates the business logic. For example, it validates that the `item` due amount is zero after a payment.

Table 25-1 contains the predefined structural and dynamic validations. Each validation is associated with a validation code.

### Table 25-1 Predefined Structural and Dynamic Validations

<table>
<thead>
<tr>
<th>Validation Type</th>
<th>Validation Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural</td>
<td>struct_valid_01</td>
<td>Validates that <code>event</code> objects point to the correct <code>item</code> objects based on the mappings configured in <code>/config/item_tags</code> and <code>/config/item_types</code>.</td>
</tr>
<tr>
<td>Structural</td>
<td>struct_valid_02</td>
<td>Validates that the PIN_FLD_PARENT field of the nonpaying child bill unit's last bill points to the parent <code>bill</code> object. Validates that the bill number of the nonpaying child bill unit and the bill number of the parent bill unit are the same.</td>
</tr>
<tr>
<td>Structural</td>
<td>struct_valid_03</td>
<td>Validates that the billing DOM of the nonpaying child bill unit and the billing DOM of the paying parent bill unit are the same.</td>
</tr>
<tr>
<td>Structural</td>
<td>struct_valid_04</td>
<td>Validates that the AR_BILLINFO_OBJ of the nonpaying child bill unit points to the <code>/billinfo</code> object of the paying parent bill unit.</td>
</tr>
<tr>
<td>Dynamic</td>
<td>dynamic_valid_01</td>
<td>Validates that the <code>item</code> due amount is zero after payment.</td>
</tr>
</tbody>
</table>

You enable these validations by setting the corresponding validation code in the ADU `pin.conf` file. ADU logs the results of the validations in the Connection Manager (CM) log file.

You can customize additional validations by modifying the `PCM_OP_ADU_POL_VALIDATE` policy opcode. See *BRM Opcode Guide*.

Configuring ADU

To configure ADU, set the entries in the ADU configuration file (`BRM_home/syssl/diagnostics/pin_adu_validate/pin.conf`) shown in Table 25-2:
<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
</table>
| `- pin_adu input_file file_name` | Set `file_name` to the name of the input file that contains the account search specification. For example:  
  `- pin_adu input_file BRM_home/sys/diagnostics/pin_adu_validate/in/input.txt` |
| `- pin_adu output_file file_name` | Set `file_name` to the name of the output folder where ADU should write the output file. For example:  
  `- pin_adu output_file BRM_home/sys/diagnostics/pin_adu_validate/out`             |
| `- pin_adu out_file_ext.ext` | Set `ext` to the output file extension. For example:  
  `- pin_adu out_file_ext.xml`                                                      |
| `- pin_adu obj_list object1; object2; ...` | Specify the objects to dump for the selected accounts. For example:  
  To dump `/billinfo` and `/payinfo` objects:  
  `- pin_adu obj_list /billinfo; /payinfo`  
  Note: Use a semicolon to separate items in the object list. |
| `- pin_adu obj_flds object1:field1, field2, ...  
  [object2:field1, field2, ...] ...` | Specify the fields in the objects to dump. For example:  
  To dump the first and last name of an account:  
  `- pin_adu obj_flds /account:PIN_FLD_NAMEINFO.PIN_FLD_FIRST_NAME, PIN_FLD_NAMEINFO.PIN_FLD_LAST_NAME`  
  To dump account invoice information:  
  `- pin_adu obj_flds /payinfo/invoice:PIN_FLD_INV_INFO`  
  Note: Use a colon to separate the items in the object list. |
| `- pin_adu dump_start_time time`  
  `- pin_adu dump_end_time time` | Set `time` in these entries to the times to use for selecting most commonly updated objects.  
  `- pin_adu dump_start_time 1175385600  
  `- pin_adu dump_end_time 1177977600`  
  Note: `time` must be in UTC format. |
| `- pin_adu struct_valid_01 n`  
  `- pin_adu struct_valid_02 n`  
  `- pin_adu struct_valid_03 n`  
  `- pin_adu struct_valid_04 n`  
  `- pin_adu dynamic_valid_01 n` | Use these entries to enable (1) or disable (0) the predefined validations. For example:  
  `- pin_adu struct_valid_01 1  
  `- pin_adu dynamic_valid_01 1` |
| `- pin_mta logfile file_name` | Set `file_name` to the ADU log file. All the debug and error messages generated by ADU are logged to this file.  
  `- pin_mta logfile BRM_home/sys/diagnostics/pin_adu_validate/adu.pinlog` |
Table 25-2  (Cont.) Entries in the ADU Configuration File

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>- pin_mta loglevel n</td>
<td>Set n to the level of information to log in the ADU log file.</td>
</tr>
<tr>
<td></td>
<td>Log level values are as follows:</td>
</tr>
<tr>
<td></td>
<td>0: no logging</td>
</tr>
<tr>
<td></td>
<td>1: log error messages only</td>
</tr>
<tr>
<td></td>
<td>2: log error messages and warnings only</td>
</tr>
<tr>
<td></td>
<td>3: log error messages, warnings, and debug messages</td>
</tr>
<tr>
<td>- pin_mta loglevel 2</td>
<td></td>
</tr>
</tbody>
</table>
Customer Management Utilities

This chapter provides reference information for Oracle Communications Billing and Revenue Management (BRM) customer management utilities.

Topics in this document:

- load_pin_business_profile
- load_pin_customer_segment
- load_pin_notify
- load_pin_verify
- load_transition_type
- pin_deferred_act
- pin_product_clear
- pin_state_change

load_pin_business_profile

Use this utility to load business profiles and validation templates into the BRM database.

See the following topics:

- About Business Profiles
- Setting Up Business Profiles and Validation Templates
- Editing the Business Profile Configuration File

Location

BRM_home/bin

Syntax


Parameters

-d
Creates a log file for debugging purposes. Use this parameter for debugging when the utility seemed to run without error but the data was not loaded into the database.

-h
Displays syntax and parameters for this utility.
-r
Retrieves the business profile and validation template data from the BRM database and saves it in an XML file.

-t
Runs the utility in test mode to validate the XML file against its schema definition. This option does not create, modify, or delete any business profile or validation template objects in your BRM database.

Note:
To avoid load errors based on XML content problems, run the utility with this option before loading data 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_business_profile other_parameters -v > filename.log
```

filename
The name and location of the business profile configuration file. The default file is BRM_home/sys/data/config/pin_business_profile.xml, but the utility can take any XML file name as a parameter as long as the file’s contents conform to the appropriate schema definition.

If you copy filename to the same directory from which you run the load utility, specify only the file name. If you run the command in a different directory from where filename is located, you must include the entire path for the file.

In addition, filename must be in the same directory as the default BRM_home/sys/data/config/business_configuration.xsd file.

Results
This utility notifies you only if it encounters errors. Look in the default.pinlog file for errors. This file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.

Note:
You must stop and restart the Connection Manager (CM) to make new business profile and validation template data available.
load_pin_customer_segment

Use this utility to load your customer segment definitions into the BRM database.

For information about customer segments, see "Creating and Managing Customer Segments".

Location

BRM_home/bin

Syntax


Parameters

-t
Runs the utility in test mode to validate the XML file format against the XSD file, and does not load the data or overwrite any existing data. Use this option before loading data into the /config object.

-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_customer_segment other_parameters –v > filename.log

-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.

-h
Displays help information for using this utility.

pin_customer_segment_file

The name and location of the XML file that stores localized strings for customer segments. The XML file is referenced by the pin_business_configuration.xsd file; therefore these files must be located in the same directory. The default customer segment file is in BRM_home/sys/data/config.

If you copy the pin_customer_segment.xml file and the pin_business_configuration.xsd file to the same directory from which you run the load_pin_customer_segment utility, you do not have to specify either the path or the file name.
If you run the command in a different directory from where the `pin_customer_segment.xml` file is located, you must include the entire path for the file.

**Results**

The `load_pin_customer_segment` utility notifies you when it successfully creates the `/config/customer_segment` object.

If the `load_pin_customer_segment` utility does not notify you that it was successful, look in the utility log file (`default.pinlog`) to find any errors. The log file is either in the directory from which the utility was started, or in a directory specified in the configuration file.

**Note:**

You must restart the Connection Manager to start recording your customer login failure preferences.

---

**load_pin_notify**

Use this utility to add an event notification list to your BRM database. See "Loading the Event Notification List" in *BRM Developer’s Guide*.

For general information about event notification, see "Using Event Notification" in *BRM Developer’s Guide*.

**Location**

`BRM_home/bin`

**Syntax**

```
load_pin_notify [-d] [-v] [-h] pin_notify_file
```

**Parameters**

- `-d`
  Specifies debug mode. The utility logs messages to the `default.pinlog` file in the current directory or in a directory specified in the utility configuration file. Use this parameter for troubleshooting when the utility runs with no errors but the notification file is not loaded into the database.

- `-v`
  Displays information about successful or failed processing as the utility runs.
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_notify other_parameters -v > filename.log
```

**-h**

Displays the syntax and parameters for this utility.

**pin_notify_file**
The name and location of the configuration file that contains the event notification list you want to load into your database. The default event notification configuration file, pin_notify, is in `BRM_home/sys/data/config`.

For more information, see "Merging Event Notification Lists" in *BRM Developer's Guide*.

**Note:**

If you copy the `pin_notify_file` to the same directory from which you run `load_pin_notify`, you do not have to specify the path or the file name.

**Results**

This utility notifies you when it successfully creates the `/config/notify` object.

If the utility does not notify you that it was successful, look in the utility log file (default.pinlog) to find any errors. The log file is either in the directory from which the utility was started or in a directory specified in the configuration file.

**Note:**

You must restart the Connection Manager to make new events and the corresponding opcodes available.

**load_pin_verify**

Use this utility to load your preferences for recording customer login failures into the BRM database.

**Location**

`BRM_home/bin`

**Syntax**

```
load_pin_verify pin_verify_file
```
Parameters

**pin_verify_file**
The name and location of the file that contains preferences for recording login failures. The default **pin_verify** file is in `BRM_home/sys/data/config`.

**Note:**
If you copy the edited **pin_verify** file to the same directory from which you run the **load_pin_verify** utility, you do not have to specify the path or the file name.

Results

The **load_pin_verify** utility notifies you when it successfully creates the `/config/verify` object.

If the **load_pin_verify** utility does not notify you that it was successful, look in the utility log file (**default.pinlog**) to find any errors. The log file is either in the directory from which the utility was started, or in a directory specified in the configuration file.

**Note:**
You must restart the Connection Manager to start recording your customer login failure preferences.

**load_transition_type**

Use this utility to load custom bundle and package transition types into the BRM database.

For information, see "Creating Custom Transition Types for Bundles and Packages".

Location

`BRM_home/bin`

Syntax

`load_transition_type [-v] [-d] [TransitionTypeFile]`

Parameters

- **-v**
  Displays information about successful or failed processing as the utility runs.

- **-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.
**TransitionTypeFile**
The name and location of the file that contains your custom transition types. By default, the utility uses `pin_transition_type`. If you run the command in a different directory from where the `pin_transition_type` file is located, you must include the entire path for the file.

**Results**
This utility notifies you only if it encounters errors. Look in the `default.pinlog` file for errors. This file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.

---

**pin_deferred_act**

Use this utility as part of your daily billing to execute deferred actions. For example, if a CSR has scheduled an account to become inactive, the `pin_deferred_act` utility performs the status change on the scheduled date. This utility is included in the `pin_bill_day` script.

See "Scheduling Status Changes in Advance".

**Location**
`BRM_home/bin`

**Syntax**

```
pin_deferred_act [-report|-purge|-retry] [-opcode opcode_name]
 [-status pending|done|error][-start mm/dd/yy]
 [-end mm/dd/yy] [-verbose] [-test] [-help]
```

**Parameters**

- **-report**
  Displays the progress and current state of `/schedule` objects. This parameter can take one or more of these options as search criteria to filter the results of your report:
  - `-opcode [opcode_name]`
  - `-status pending|done|error`
  - `-start mm/dd/yy` or `yyyy`
  - `-end mm/dd/yy` or `yyyy`

  For example:

  ```
  pin_deferred_act -report -start 01/10/03 -end 01/24/03 -verbose
  ```

- **-purge**
  Purges from the Oracle Communications Billing and Revenue Management (BRM) database all `/schedule` objects whose actions have been executed successfully. This helps reduce the size of your database. This parameter can take one or more of these options as search criteria for purging:
  - `-opcode [opcode_name]`
For example:

```
pin_deferred_act -purge -start 01/10/03 -end 01/24/03 -verbose -opcode PCM_OP_BILL_MAKE_BILL_NOW
```

-retry
Retries all the /schedule objects whose schedule actions have failed to execute and whose status is marked as ERROR. This parameter can take one or more of these options as search criteria for purging:

- **-opcode [opcode_name]**
- **-status pending|done|error**
- **-start mm/dd/yy or yyyy**
- **-end mm/dd/yy or yyyy**

For example:

```
pin_deferred_act -retry -start 01/10/03 -end 01/24/03 -verbose
```

-opcode [opcode_name]
Used as search criteria by the -report, -purge, and -retry parameters for retrieving /schedule objects containing the specified opcode responsible for the deferred action. For example:

```
pin_deferred_act -report -start 01/10/03 -end 01/24/03 -verbose -opcode PCM_OP_BILL_MAKE_BILL_NOW
```

-status pending|done|error
Used as search criteria by the -report, -purge, and -retry parameters for retrieving /schedule objects having the specified status. For example:

```
pin_deferred_act -retry -start 01/10/03 -end 01/24/03 -verbose -status ERROR
```

-start mm/dd/yy or yyyy
-end mm/dd/yy or yyyy
Used as search criteria by the -report, -purge, and -retry parameters for retrieving /schedule objects with an execution date matching the start and end dates specified. The value you supply for the start date is inclusive, but the value for the end date is non-inclusive and also defaults to the current date. If a start date is not specified, this utility retrieves all valid /schedule objects up to the specified end date. If an end date is not specified, this utility uses the current date as the end date and retrieves all valid /schedule objects until the current date.

-verbose | -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:

```
pin_deferred_act any_other_parameter -v > filename.log
```

**-test**
Runs a test to find out how many accounts meet the criteria without performing the action. The test has no effect on the accounts. This is most useful when run with the `-verbose` option.

**-help | -h**
Displays the syntax and parameters for this utility.

**Results**

If the utility does not notify you that it was successful, look in the utility log file (`default.pinlog`) to find any errors. The log file is either in the directory from which the utility was started, or in a directory specified in the configuration file.

When it is called internally by the `pin_bill_day` script, the `pin_deferred_act` utility logs error information in the `pin_mta.pinlog` file.

---

**pin_product_clear**

Use this utility to purge the Audit Accounts Product table of data that is older than a specified time period. When you modify a charge offer, BRM records the details of the charge offer modifications in the `AUDIT_ACCOUNTS_PRODUCT_T` table. BRM stores audit data when a customer purchases an item charge offer and modifies or cancels an existing subscription charge offer.

**Location**

`BRM_home/bin`

The `pin.conf` file for this utility is located in `BRM_home/apps/pinapps/pin_rerate`. Run `pin_product_clear` from this directory.

**Syntax**

```
pin_product_clear  -n days  -d mm/dd/yyyy  [-verbose]  [test]  
[-help]
```

**Parameters**

**-n days**
Deletes all the records from the Audit Accounts Product table that are older than the number of days specified by days.

**-d mm/dd/yyyy**
Deletes all records from the Audit Account Product table that are dated prior to the midnight of the date you have specified.
-verbose
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 `verbose` 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_product_clear other_parameters -v > filename.log
```

-test
Tests the utility, but does not affect the Audit Accounts Product table.

-help
Displays the syntax and parameters for this utility.

**Results**
If the `pin_product_clear` utility doesn't notify you that it was successful, look in the utility log file (`pin_product_clear.pinlog`) to find any errors. The log file is either in the directory from which the utility was started, or in a directory specified in the configuration file.

**pin_state_change**
Use this utility to perform bulk service state transitions based on the state expiration time. See "Creating Custom Service Life Cycles".

**Location**
`BRM_home/bin`

The configuration file for this utility is located in the `BRM_home/apps/pin_state_change` directory. Run `pin_state_change` from that directory.

**Syntax**
```
pin_state_change [-help] [-state] [-service] [-verbose]
```

**Parameters**
- **-help**
  Displays the syntax and parameters for this utility.

- **-state**
  Performs state transitions for services in this state only. Use the name of a state defined in the `<NAME>` element in a `config_lifecycle_states.xml` file.

- **-service**
  Performs state transitions for this service type only. Use the name of a BRM service type (`/service/*`) that uses a custom life cycle.
-verbose
Displays information about successful or failed processing as the utility runs.

Results
The log file specified in the utility's configuration (pin.conf) file records the following:

- The number of services for which this utility made a state change
- The number of state changes that were successful