## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>vii</td>
</tr>
<tr>
<td>Audience</td>
<td>vii</td>
</tr>
<tr>
<td>Downloading Oracle Communications Documentation</td>
<td>vii</td>
</tr>
<tr>
<td>Documentation Accessibility</td>
<td>vii</td>
</tr>
</tbody>
</table>

1 List of Accounts Receivable Features

Accounts Receivable Features ............................................................................................................. 1-1

2 Making Adjustments

About Adjustments ........................................................................................................................................ 2-1

3 Adjusting Multiple Accounts Simultaneously

Adjusting Multiple Accounts Simultaneously ............................................................................................. 3-1

4 Opening and Resolving Disputes

About Disputes and Settlements .................................................................................................................... 4-1
Reserving and Freeing Balances for Disputes and Settlements ......................................................................... 4-3
Configuring Event Notification for Disputes and Settlements ........................................................................ 4-3

5 Configuring Reasons for Adjustments, Disputes, and Settlements

Configuring Reason Codes for Adjustments, Disputes, and Settlements .......................................................... 5-1

6 Giving Refunds to Customers

About Refunds ................................................................................................................................................... 6-1
Creating Refund Items for All Accounts with a Credit Balance ........................................................................ 6-2
Giving Refunds for BRM-Initiated Payments ..................................................................................................... 6-2
Giving Refunds for Externally Initiated Payments ............................................................................................ 6-2
Specifying the Minimum Amount to Refund ....................................................................................................... 6-3
Defining Nonrefundable Items .......................................................................................................................... 6-3
Reversing Refunds ............................................................................................................................................. 6-3

7 Configuring Write-Offs

About Write-Offs ............................................................................................................................................... 7-1
8 Configuring Write-Off Reversals

About Write-Off Reversals ........................................................................................................... 8-1
About Overpayment and Underpayment Allocation to a Written-off Account or Bill .......... 8-2
Defining Reason Codes for Write-Off Reversals ....................................................................... 8-3
Mapping G/L IDs to Write-Off Reversal Events ....................................................................... 8-3
Enabling Automatic Write-Off Reversals during Payment Collection .................................. 8-4
Enabling Write-Off Reversals for Bill Units .............................................................................. 8-4

9 Creating and Managing Account and Bill Unit Hierarchies

About Account Groups .................................................................................................................. 9-1
  About Bill Units and Account Groups ...................................................................................... 9-2
  Comparing Hierarchy and Sharing .......................................................................................... 9-2
  Hierarchy Balance Impacts ...................................................................................................... 9-4
  Sharing Group Balance Impacts .............................................................................................. 9-5
About Account Hierarchies ......................................................................................................... 9-5
  Performance Impact of Account Hierarchies ........................................................................ 9-6
About Bill Unit Hierarchies ........................................................................................................ 9-6
  How Account Status Changes Affect Hierarchies ................................................................. 9-11
  Currency Requirements of Hierarchies ................................................................................ 9-12
  Billing Setups in Hierarchies ................................................................................................. 9-12
Calculating Balance Due in Account and Bill Unit Hierarchies ............................................ 9-12
  Who Pays for Open Items and Pending Items? .................................................................... 9-14
  Multiple Levels of Parent Accounts .................................................................................... 9-15
  Hierarchy Changes and Billing Dates .................................................................................... 9-16
  Examples of Changes to Account Hierarchies ..................................................................... 9-17
Creating Account and Bill Unit Hierarchies .......................................................................... 9-19
Managing Account Hierarchies ................................................................................................. 9-19
  Moving Closed Accounts into or out of Hierarchies ............................................................ 9-19

10 Creating and Managing Charge and Discount Sharing Groups

About Charge and Discount Sharing Groups ............................................................................. 10-1
  Working with Complex Charge and Discount Sharing Groups ......................................... 10-2
About Discount Sharing Groups ................................................................................................ 10-3
  How Account Status Changes Affect Discount Sharing Groups ....................................... 10-4
  How Group Owner Changes Affect Discount Sharing Groups ........................................... 10-4
  Members and Discount Sharing Groups .............................................................................. 10-5
  Currency Requirements of Discount Sharing Groups ......................................................... 10-6
  Billing for Discount Sharing Groups ..................................................................................... 10-6
  Configuring the Start and End Times for Discount Sharing ................................................ 10-6
About Charge Sharing Groups .................................................................................................. 10-7
  About Charge Sharing Group Owners .................................................................................. 10-8
    How Owner Account Status Changes Affect Charge Sharing ........................................ 10-8
    About Changing Charge Sharing Group Owners ............................................................. 10-9
  About Charge Sharing Group Members .............................................................................. 10-9
  Currency Requirements for a Charge Sharing Group .......................................................... 10-10
  Billing for a Charge Sharing Group ...................................................................................... 10-10
About Global Charge Sharing Groups ................................................................. 10-11
About the Order in Which Charges Are Applied to Groups ................................. 10-11
About Creating Global Charge Sharing Groups ................................................. 10-11
Enabling Global Charge Sharing Searches during Discounting ........................... 10-12
Using Third-Party Applications to Manage Global Charge Sharing Groups ........... 10-12
About Creating, Modifying, and Deleting Charge and Discount Sharing Groups ...... 10-13
About Creating Charge and Discount Sharing Groups ........................................ 10-14
About Modifying Charge and Discount Sharing Groups ...................................... 10-14
About Deleting Charge and Discount Sharing Groups ........................................ 10-16
How Charges and Discounts Are Applied .......................................................... 10-16
About Ordered Balance Groups ........................................................................... 10-16
How Discounts Are Applied When a Member Belongs to the Group Owner Account ... 10-19
Creating, Deleting, and Modifying Ordered Balance Groups ............................... 10-19
Creating Ordered Balance Groups ....................................................................... 10-19
Deleting Ordered Balance Groups ....................................................................... 10-19
Modifying Ordered Balance Groups .................................................................... 10-19
Modifying the Order in Which Charge and Discount Sharing Groups Are Used .... 10-20
Creating or Modifying Multiple Ordered Balance Groups Simultaneously .......... 10-20

11 About Transferring Rollover Balances

About Allowing Customers to Transfer Rolled-Over Balances ............................... 11-1
About the Rollover-Transfer Profile .................................................................... 11-2
About Rerating the Receiver’s Account due to Delayed Billing ............................ 11-2
Configuring BRM to Use Rollover Transfers ....................................................... 11-2

12 Managing Balances

Transferring Balance Amounts ............................................................................. 12-1
Moving a Balance Group from One Bill Unit to Another ........................................ 12-1
Backdating A/R Actions ....................................................................................... 12-2
Synchronizing BRM and ECE Balance Group Transfer Data ............................... 12-3

13 Balance Monitoring

About Balance Monitoring ..................................................................................... 13-1
About Monitor Groups ......................................................................................... 13-1
About Monitor Group Owners ............................................................................. 13-2
About Monitor Group Members .......................................................................... 13-2
About Monitor Group Types .............................................................................. 13-2
About the Balances of a Monitor Group ............................................................. 13-3
Balance Impacts Included in a Monitored Balance ............................................. 13-3
About Monitoring Charge Sharing and Sponsored Account Balances .................. 13-3
When Balance Impacts Are Added to a Monitored Balance ............................... 13-3
Alerting Customers When Monitored Balances Cross Limits or Thresholds ........ 13-4
About Setting Limits and Thresholds .................................................................... 13-4
About Using Event Notification to Alert Customers ........................................... 13-5
About Notification Events for Balance Monitoring ............................................. 13-5
Providing Real-Time Access to the Balances of Monitor Groups ......................... 13-6
About Creating and Maintaining Balance Monitors ................................................................. 13-6
Balance Monitor Creation Process Overview ........................................................................... 13-7
About Using AMS to Manage Balance Monitors Automatically ............................................ 13-7
Managing Balance Monitors without AMS ........................................................................... 13-8
Balance Monitoring Process Overview ................................................................................. 13-9
About Balance Monitoring and Real-Time Rating ................................................................. 13-9
Understanding the Monitor Queue ....................................................................................... 13-9
Using pin_monitor_balance to Update Monitored Balances ................................................. 13-10
Configuring Your BRM System for Balance Monitoring ........................................................ 13-10
Enabling Balance Monitoring in BRM ................................................................................... 13-11
Enabling AMS in BRM ......................................................................................................... 13-12
Configuring Event Notification for Balance Monitoring ...................................................... 13-13
Specifying a Wait Time before Rerating Events .................................................................. 13-14
Configuring pin_monitor_balance to Process Events in the Order Created ....................... 13-14
Running pin_monitor_balance to Update Monitored Balances .............................................. 13-15
Specifying Whether Item Transfers Affect Balance Monitors ............................................... 13-15
Implementing Balance Monitoring in Custom Client Applications ........................................ 13-15
About Implementing Balance Monitoring in Client Applications ........................................... 13-16
Creating and Maintaining Balance Monitor Objects .............................................................. 13-16
Using AMS to Create and Maintain Balance Monitors Automatically ................................... 13-17
Using the Balance Monitor API to Create and Maintain Balance Monitors ......................... 13-17
Creating /balance_group/monitor Objects .......................................................................... 13-18
Modifying /balance_group/monitor Objects ....................................................................... 13-19
Deleting /balance_group/monitor Objects ........................................................................... 13-19
Creating, Modifying, or Deleting /group/sharing/monitor Objects ....................................... 13-19
Changing the Owner of a Balance Monitor .......................................................................... 13-20
Validating the Members of a Balance Monitor Group ......................................................... 13-21
Adding a Monitor Group to a Member’s /ordered_balgrp Object .......................................... 13-21
Adding and Removing Balance Monitor Members Automatically ......................................... 13-22
Adding Members to Newly Created Balance Monitors Automatically .................................. 13-22
Adding Members to Hierarchy-Type Monitors ...................................................................... 13-23
Adding Members to Payment Responsibility-Type Monitors ............................................... 13-24
Updating Hierarchy-Type Monitors Automatically ............................................................... 13-24
Updating Paying Responsibility-Type Monitors Automatically ............................................ 13-25
Updating Subscription-Type Monitors Automatically ............................................................ 13-26
Removing Members from Hierarchy- and Paying Responsibility-Type Monitors ................... 13-26
Displaying Balance Monitor Information in Client Applications ............................................ 13-27
Retrieving the Balances for a Monitor Group ...................................................................... 13-27
Retrieving the Balance Monitors Owned by an Account or Service ..................................... 13-27
Updating Monitor Balances and Sending Credit Limit/Threshold Breach Notifications .......... 13-28
Example of Credit Threshold Notification Event Generation ................................................ 13-29

14 Accounts Receivable Utilities

pin_apply_bulk_adjustment ....................................................................................................... 14-2
pin_monitor_balance ............................................................................................................. 14-3
pin_mass_refund ..................................................................................................................... 14-4
pin_refund .............................................................................................................................. 14-5
This guide describes how to use and manage accounts receivable (A/R) data in Oracle Communications Billing and Revenue Management (BRM).

Audience

This guide is intended for operations personnel 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


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

To learn about accounts receivable, see BRM Concepts.

## Accounts Receivable Features

This table lists accounts receivable features.

<table>
<thead>
<tr>
<th>Task</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make an adjustment to a customer’s balance</td>
<td>Making Adjustments</td>
</tr>
<tr>
<td>Open or resolve a disputed amount in a customer’s balance</td>
<td>Opening and Resolving Disputes</td>
</tr>
<tr>
<td>Transfer a balance between accounts</td>
<td>Transferring Balance Amounts</td>
</tr>
<tr>
<td>Change the reasons that are displayed when an adjustment, dispute, or settlement is made</td>
<td>Configuring Reasons for Adjustments, Disputes, and Settlements</td>
</tr>
<tr>
<td>Write off customer debt</td>
<td>Configuring Write-Offs</td>
</tr>
<tr>
<td>Reverse the write-off of customer debt</td>
<td>Configuring Write-Off Reversals</td>
</tr>
<tr>
<td>Give a refund to a customer</td>
<td>Giving Refunds to Customers, Specifying the Minimum Amount to Refund, Defining Nonrefundable Items</td>
</tr>
<tr>
<td>Reverse a refund</td>
<td>Reversing Refunds</td>
</tr>
<tr>
<td>Create and manage account groups</td>
<td>Creating and Managing Account and Bill Unit Hierarchies</td>
</tr>
<tr>
<td>Manage bill unit hierarchies to define paying and non-paying bill units</td>
<td>Creating and Managing Charge and Discount Sharing Groups</td>
</tr>
<tr>
<td>Create and manage groups of accounts that share charges and discounts</td>
<td>About Transferring Rollover Balances</td>
</tr>
<tr>
<td>Manage how customers are allowed to transfer rollover balances</td>
<td>Managing Balances</td>
</tr>
<tr>
<td>Manage customer balances</td>
<td>Moving a Balance Group from One Bill Unit to Another</td>
</tr>
<tr>
<td></td>
<td>Backdating A/R Actions</td>
</tr>
<tr>
<td></td>
<td>Synchronizing BRM and ECE Balance Group Transfer Data</td>
</tr>
</tbody>
</table>
Making Adjustments

This document provides an overview of how adjustments are performed in Oracle Communications Billing and Revenue Management (BRM).

See also:
- Adjusting Multiple Accounts Simultaneously.
- List of Accounts Receivable Features.
- BRM Concepts.

To reverse an adjustment, use PCM_OP_AR_REVERSE_ADJUSTMENT. See BRM Opcode Guide.

About Adjustments

An adjustment is a transaction that debits or credits a customer’s account by changing the amount due for a bill item, or the amount of a noncurrency balance.

- A credit adjustment decreases the customer’s balance; that is, it decreases the amount a customer owes. A credit adjustment is represented as a negative number. For example, when you give 100 free minutes, the adjustment is represented as -100.

- A debit adjustment increases the customer’s balance.

Customer service representatives perform adjustments on a variety of levels. For example:

- Event level: If the customer made a 10-minute call that was mistakenly billed as a 30-minute call, CSRs perform the adjustment for that specific call at the event level.

- Account level: If the customer’s package provided 100 included minutes a month, but charges started accruing after only 30 minutes, CSRs perform the adjustment at the account level instead.

Credit adjustments do the following:

- For currency balances, decrease the Due of a bill item
- For noncurrency balances, increase the adjusted balance

If a CSR is crediting an event, the balance impact of that event is removed from the customer’s account. Debit adjustments have the opposite effect.

The way that BRM processes adjustments and records the adjustment’s balance impact varies from level to level, as follows:
About Adjustments

■ A/R and individual account: Adjustments at this level reduce the current balance of the customer’s bill. The account’s default balance group is decreased by the amount of the credit.

■ Subscription service and member service: Adjustments at these levels are similar to account adjustments. However, the adjustment targets a specific balance group associated with the subscription service or member service rather than using only the default balance group. In this case, BRM uses the balance group supplied by the /service object associated with the subscription service or member service selected by the CSR.

As with account adjustments, the CSR must allocate the adjustment before it affects the customer’s bill.

■ Bill level: Adjustments at this level reduce the current balance of the customer’s bill. Here, the amount of the adjustment is subtracted from the due amount for the bill, and payment for that amount is not requested.

Adjustments can be made to an entire bill or a selection of bill items, distributing the adjustment as a fixed amount per item or as a percentage. In either case, BRM creates a single adjustment item and transfers the credit to the bill items covered by the adjustment. The current balance of the appropriate balance groups is reduced by the amount of the credit.

Bill adjustments act against A/R bills only. CSRs cannot adjust a bill from a nonpaying bill unit (/billinfo object) by filing a bill adjustment directly against that bill. Instead, they file the bill adjustment against the parent A/R bill. Also, the adjustment amount cannot exceed the total amount of the bill against which the adjustment is applied.

■ Item level: When you adjust a bill item, the amount of the adjustment is subtracted from the Due of the bill item, and payment for that amount is not requested. The current balance of the appropriate balance group is reduced by the amount of the credit.

■ Event level: Adjustments at this level depend on whether the adjustment occurs before billing or after billing. In either case, the original event is never adjusted.

– If the adjustment occurs before billing has run, it changes the balance impact of the shadow event.

– If the adjustment occurs after billing has run, it changes the balance impact of the adjustment event (/event/billing/adjustment/event).

Table 2–1 summarizes how adjustments handle currency and noncurrency balances.

<table>
<thead>
<tr>
<th>Adjustment Type</th>
<th>Currency</th>
<th>Noncurrency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account adjustment¹</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Subscription service adjustment ¹</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Member service adjustment ¹</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Bill adjustment</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Item adjustment</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Event adjustment</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

¹ For these adjustments, there must be a noncurrency balance group at the account level for the adjustment to affect a noncurrency balance.
An event adjustment that credits currency reduces the balance impact of the event and its General Ledger (G/L) impact. It does not cancel the event, only the cost of the event.

**Important:** When adjusting pending items, ensure proper G/L reporting by specifying that BRM create a shadow adjustment instead of a standard adjustment. See the discussion of the PCM_OP_AR_EVENT_ADJUSTMENT opcode in *BRM Opcode Guide*. 
Adjusting Multiple Accounts Simultaneously

This document describes how to make a bulk adjustment in Oracle Communications Billing and Revenue Management.

See also:
- Making Adjustments.
- List of Accounts Receivable Features.

Adjusting Multiple Accounts Simultaneously

In some situations, you may want to adjust multiple paying accounts simultaneously. For example, if regular daily rates were mistakenly applied during a holiday discount period, you would most likely want to perform adjustments for all the accounts that used the service over the holiday.

This type of adjustment is called a bulk adjustment. To perform a bulk adjustment, you create a file that lists each of the accounts that should receive the bulk adjustment and provides information about the adjustment. You then run the `pin_apply_bulk_adjustment` utility to perform the adjustments based on the data in the file.

The file also specifies whether the adjustment is taxable. BRM bases tax calculation on the full adjustment amount for each account. Depending on how adjustment taxation is configured, BRM may apply the adjustment tax reversal when the bulk adjustment runs or send the adjustment amount for deferred taxation during the next billing run.

When BRM processes a bulk adjustment, it performs each adjustment as a separate transaction. Therefore, there is no need to roll back the successful adjustments if some adjustments fail. BRM reports any accounts that failed the bulk adjustment in a log file. All the CSV records that failed to process due to wrong format or due to the server side errors are written to an error file. The default name of this file is `pin_mta_failed.csv`. The names and locations of these files can be defined in the `pin_apply_bulk_adjustment pin.conf` file.

The file is in CSV format and is typically supplied by an external source program. CSV is a standard file format that uses commas as field delimiters and line breaks as record delimiters. CSV files should contain no blank lines. CSV files used for bulk adjustments are typically generated by an external program, and the records must use the following format:

```
account_POID, adjustment_amount, balance_group_POID, tax_flag, tax_code, tax_supplier_ID, balance_element_ID, end_time, reason_code_domain, reason_code, description
```
Consider these guidelines:

- The account_POID, adjustment_amount, and balance_element_ID fields are mandatory. You can omit any of the other fields. If you omit any field in a record (including those at the end of the record), you must still include the associated comma so BRM can keep track of which field it is processing.
- If you omit the balance_group_POID, BRM adjusts the default balance group.
- The tax_flag field indicates whether the adjustment requires tax reversal. If the flag is set to 1, the adjustment does not need tax reversal. If the flag is set to 2, the adjustment includes a tax reversal. If you omit this field, the adjustment occurs without a tax reversal.
- The balance_element_ID is the numeric code for the balance element you are adjusting (for example, 840 for US dollars). You can specify currency or noncurrency balance elements.
- The format for the end_time field is MM/DD/YYYY. BRM uses midnight for the date you specify as the time stamp for the adjustment.
- If you include a reason_code_domain, you must also include a reason_code, and the reverse.
- If the description field includes any commas, you must enclose the field in quotation marks ("'). Quotation marks are optional for description fields that do not include commas.

The following CSV file segment shows a bulk adjustment with tax:

```
0.0.0.1 /account 15269 0, -9.5, 0.0.0.1 /balance_group 12901 0, 2, , , 840, , , , Rate issue
0.0.0.1 /account 12581 0, -9.5, 0.0.0.1 /balance_group 16165 0, 1, , , 840, , , , Rate issue
0.0.0.1 /account 15557 0, 10, , , , , 1000010, 04/26/2004, 12, 5, "Service drop, fix this"
```

In the preceding example, the first record specifies an adjustment with a tax reversal against the currency balance element for a specific balance group. The adjustment amount is $9.50. The second record specifies a similar adjustment, except that this adjustment has no tax reversal. The last record specifies an adjustment of 10 free domestic minutes to the account’s default balance group. BRM does not apply a tax reversal for this adjustment.

To run bulk adjustments, generate a CSV file and then run the "pin_apply_bulk_adjustment" utility to load the file into the database.

1. Generate the CSV file and check it for format problems and spurious blank lines.
2. Use the following command to run the pin_apply_bulk_adjustment utility:
   ```
   pin_apply_bulk_adjustment -f input_file.csv
   ```

   **Important:** If the CSV file is not in your working directory, use the full path to the file:
   ```
   pin_apply_bulk_adjustment -f /files/bulk_adjust/input_file.csv
   ```
3. Check the pin_bulk_adjust.pin log file for any failed adjustments.
The bulk adjustment application generates an intermediate file in flist format. The default name of this file is `pin_mta_search.flist`.

All the CSV records that failed to process due to wrong format or due to the server side errors are written to an error file. The default name of this file is `pin_mta_failed.csv`.

The names and locations of these files can be defined in the `pin_apply_bulk_adjustment pin.conf` file.
Adjusting Multiple Accounts Simultaneously

3-4 Managing Accounts Receivable
Opening and Resolving Disputes

This document describes how to open and resolve disputes in Oracle Communications Billing and Revenue Management.

Topics in this document:

- About Disputes and Settlements
- Reserving and Freeing Balances for Disputes and Settlements
- Configuring Event Notification for Disputes and Settlements

See also:

- Making Adjustments.
- List of Accounts Receivable Features.

About Disputes and Settlements

A CSR creates a dispute when a customer disagrees with the amount he or she is asked to pay and the problem requires investigation before it can be resolved. Disputes and settlements credit or debit the currency or noncurrency balances of a customer’s account but do not return money to the customer directly.

The dispute process involves two distinct activities: opening a dispute and settling that dispute. A dispute is a transaction that records a customer’s objection to a currency amount billed to an account. A settlement is a transaction that resolves a dispute by crediting or debiting all, part, or none of the dispute amount to the account.

As with payments and adjustments, the BRM system creates a dispute item when the CSR enters a dispute. The disputed amount typically reduces the Due of a bill and the current balances of the customer’s account. BRM does not request payment for the disputed amount. If an item is under dispute (that is, the disputed field is nonzero) the bill item remains open even if the Due becomes zero.

In settling a dispute, the CSR can grant the entire disputed amount to the customer, grant only part of the disputed amount, or deny the dispute and again request payment for the entire disputed amount. The BRM system creates a settlement item for the amount that is not granted and transfers amounts between the settlement item and the disputed bill item. After the dispute resolution, the Due and the current balance in the customer’s account balance group reflect what the customer owes.

When the dispute item is created, its Total and Due are initially equal, but the BRM system immediately transfers the credit in the Due field of the dispute item to the disputed bill item, makes the Due of the dispute item zero, and closes the dispute item.
After a settlement, the amount of the dispute that is granted to the customer appears in the Adjusted field of the disputed bill item and in the histories of the disputed item and the settlement item.

For currency balances, credit disputes decrease the Due of a bill or bill item. If the customer is disputing the currency balance of an event for credit, the dispute removes the balance impact of the disputed event from the account. Debit disputes have the opposite effect.

---

**Important:** For noncurrency balances, event disputes are recorded but do not directly affect the balance until the dispute is settled. At that point, the disputed balance is increased by the amount of the settlement.

---

Disputes and settlements are made at various levels, as appropriate to the situation. The way that BRM processes disputes and settlements varies slightly from level to level, as follows:

- **Bill level:** CSRs can dispute or settle an entire bill or a selection of bill items. In either case, BRM creates a single dispute or settlement item and alters the Due of each bill item covered by the dispute or settlement. Bill disputes and settlements can be thought of as a set of item disputes or settlements covered under the umbrella of a single dispute or settlement item.

  Bill disputes and settlements can act against A/R bills only. CSRs cannot dispute or settle a bill from a nonpaying bill unit by filing a bill dispute or settlement directly against that bill. Instead, they file the bill dispute or settlement against the parent A/R bill. The dispute amount cannot exceed the total amount of the bill against which the dispute is applied, and the settlement amount cannot exceed the total dispute amount for the bill.

- **Item level:** Disputes and settlements at this level operate as either a CSR-initiated action against a single bill item or as a unilateral action against a set of bill items initiated by a bill dispute or settlement. Depending on how the dispute or settlement was initiated, BRM creates dispute and settlement items as follows:
  - If the CSR initiated the dispute or settlement at the item level, one dispute or settlement item is created for the item he or she chooses.
  - If the dispute or settlement was initiated at the bill level, a single dispute or settlement item will cover all the individual item disputes that make up the bill dispute.

To customize item level disputes, use the PCM_OP_BILL_POL_VALID_DISPUTE and PCM_OP_BILL_POL_VALID_SETTLEMENT opcodes. See *BRM Opcode Guide*.

- **Event level:** CSRs can dispute and settle any event and they can dispute or settle multiple events from an account in one operation.

  When an event dispute is opened, BRM creates one dispute event for each disputed event, establishing a one-to-one correspondence between the dispute event and the original event. The dispute event updates the original item’s Dispute field. BRM bundles all the individual dispute events into one dispute item.

  Similarly, when you settle an event dispute, BRM creates a settlement event for each dispute event. In this case, BRM transfers the settlement amount to the Adjusted field of the original item and the denied amount to the Due field. As
with disputes, BRM bundles all individual settlement events into one settlement item.

Table 4–1 summarizes the dispute and settlement types.

<table>
<thead>
<tr>
<th>Dispute or Settlement Type</th>
<th>Currency</th>
<th>Noncurrency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bill dispute or settlement</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Item dispute or settlement</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Event dispute or settlement</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

CSRs open and resolve disputes by using Billing Care.

**Reserving and Freeing Balances for Disputes and Settlements**

To ensure that balances are not misused when a dispute is open, BRM reserves balances disputed at the event level. The reservation process protects the account balance from being credited for the dispute amount until settlement occurs.

For example, if the customer disputed a 10-minute charge on his bill, this amount would be credited to the account balance and, for prepaid accounts, would increase the prepaid amount. However, placing a reservation on that 10 minutes lowers the credit limit for the account and prevents the disputed time from being freely used as it would be in a normal credit situation.

**Configuring Event Notification for Disputes and Settlements**

When an event dispute is opened or settled, BRM uses event notification to call opcodes that perform the appropriate follow-up operations. Disputes and settlements use the following events for event notification:

- `/event/billing/dispute/notify`
- `/event/billing/settlement/notify`

Before you can use the event dispute and settlement feature, you must configure the event notification feature as follows:

1. If your system has multiple configuration files for event notification, merge them. See "Merging Event Notification Lists" in BRM Developer’s Guide.

2. Ensure that the merged file includes the following information from the BRM_home/sys/data/config/pin_notify file:

```plaintext
# Settlements and disputes related event notifications
2354 0 /event/billing/settlement/notify
2355 0 /event/billing/dispute/notify
```

3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list. See "Editing the Event Notification List" in BRM Developer’s Guide.

4. (Optional) If necessary to accommodate your business needs, create custom code for event notification to trigger. See "Triggering Custom Operations" in BRM Developer’s Guide.
5. Load your final event notification list into the BRM database.
   
   See "Loading the Event Notification List" in BRM Developer’s Guide.
   
   For more information, see "Using Event Notification" in BRM Developer’s Guide.
Configuring Reasons for Adjustments, Disputes, and Settlements

When a CSR enters an adjustment, dispute, or settlement, the CSR enters a reason for the action; for example, "Net Speed Was Slow." You define those reasons in reason codes.

This document describes how to configure reason codes for accounts receivable actions in Oracle Communications Billing and Revenue Management.

See also:
- Making Adjustments
- Opening and Resolving Disputes
- List of Accounts Receivable Features

Configuring Reason Codes for Adjustments, Disputes, and Settlements

Reason codes explain why an adjustment, dispute, or settlement is being granted. You typically select from a list of reasons appropriate to the action.

To implement reason codes, you create and load a reasons.locale file that lists each reason code and associates it with a reason code domain. The domain classifies the reason code according to the type of activity it applies to. You can also include information in the reasons.locale file that prevents inappropriate event adjustments based on reason code and service.

The reasons.locale file defines each reason code domain, the reason codes that belong to the domain, and the event G/L ID. The domain and reason code information is used to build the /strings object and the event G/L ID is used to build the /config/map_glid object.

For event adjustments, you can also use the service block to define which services are valid for the reason code and specify the valid event types within those services. The service block is optional; it is used to define the reason code scope. If you do not include it, BRM adjusts all events in the batch regardless of whether the reason for the adjustment makes sense for the service and event.

The following example shows a single reasons.locale file segment defining a reason code domain.

```
DOMAIN = 'Reason Codes - Event Adjustment Reasons';
STR
  ID = 3;
  VERSION = 22;
  STRING = 'Net Speed Was Slow';
```
For more information on the `reasons.locale` file, see "String Manipulation Functions" in BRM Developer’s Reference.

To define reason codes for adjustments, disputes and settlements, you edit the `reasons.en_US` sample file in the `BRM_home/sys/msgs/reasoncodes` directory. You then use the `load_localized_strings` utility (see BRM Developer’s Guide) to load the contents of the file into the `/strings`, `/config/map_glid`, and `/config/reason_code_scope` objects.

When you run the `load_localized_strings` utility, use this command:

```
load_localized_strings reasons.locale
```

---

**Note:**

- If you’re loading a localized version of this file, use the correct file extension for your locale. For a list of file extensions, see "Locale Names" in BRM Developer’s Guide.
- If you add your own reason codes to the `reasons.locale` file, you should use IDs above 100,000.

---

**Caution:** The `load_localized_strings` utility overwrites existing `/config/map_glid` and `/config/reason_code_scope` objects. If you are updating these objects, you cannot load new G/L ID maps and reason code scopes only. You must load complete sets of data each time you run the `load_localized_strings` utility. This is also true when using the `/strings` object, but only if you specify the `-f` parameter. Otherwise, the `load_localized_strings` utility appends the new data to the object.

For information on loading the `reasons.locale` file, see "Loading Localized or Customized Strings" in BRM Developer’s Guide. For information on creating new strings for this file, see "Creating New Strings and Customizing Existing Strings" in BRM Developer’s Guide.
Giving Refunds to Customers

This document describes how to give refunds in Oracle Communications Billing and Revenue Management (BRM).

Topics in this document:
- About Refunds
- Creating Refund Items for All Accounts with a Credit Balance
- Giving Refunds for BRM-Initiated Payments
- Giving Refunds for Externally Initiated Payments
- Specifying the Minimum Amount to Refund
- Defining Nonrefundable Items
- Reversing Refunds

See also:
- Making Adjustments
- Opening and Resolving Disputes
- List of Accounts Receivable Features

About Refunds

You can give customers a refund whenever your company owes them money. Unlike an adjustment, which credits the customer’s account balances, a refund returns money that your company owes a customer directly to the customer.

Refunds for BRM-initiated payments return money to the direct debit or credit card account the customer uses to make payments. For customers who pay invoices, your company makes a refund by check or other externally initiated payment method.

Note: You cannot refund suspended payments.

Giving refunds to customers is a two-step process. You first create refunds as refund items in the BRM database. You then run the `pin_refund` utility to deposit the refunds and close the refund items.

There are two ways to create refund items:
- Use Billing Care to create refund items for individual accounts. You can create refunds for accounts or bills that have a credit balance.
Creating Refund Items for All Accounts with a Credit Balance

- Use the **pin_mass_refund** utility to create refund items for all accounts that have a credit balance (that is, all accounts that your company owes money to).

When creating a refund item, the BRM evaluates the open bill items and A/R action items for the account, transfers credit among the items to close as many as possible, creates a refund item to contain the remaining credit, and closes the remaining open items that had a credit in Due.

When you refund an account whose bill unit is nonpaying, the refund is applied to the account that owns the paying parent bill unit. BRM associates the refund item with the parent bill unit’s balance group that has the greatest amount due. If the parent bill unit has no amount due, the refund item is associated with the default balance group of the account that owns the parent bill unit. For more information about bill unit hierarchies, see "Creating and Managing Account and Bill Unit Hierarchies".

Creating Refund Items for All Accounts with a Credit Balance

Run the **pin_mass_refund** utility to create a refund item for accounts that have a credit balance. See "**pin_mass_refund**".

Include the **pin_mass_refund** utility in any of the billing scripts, such as the **pin_bill_day** script. Run the **pin_mass_refund** utility before you run the **pin_refund** utility. See "Running Billing Utilities" in **BRM Configuring and Running Billing**.

Giving Refunds for BRM-Initiated Payments

Run the **pin_refund** utility to give refunds to all accounts that have open refund items. See "**pin_refund**".

You do not specify start and end date parameters when you run this utility. If you miss a billing day, the **pin_refund** utility processes on all existing refund items.

The **pin_refund** utility is included by default in the **pin_bill_day**, **pin_bill_week**, and **pin_bill_month** scripts. For information about the billing scripts, see **BRM Configuring and Running Billing**.

---

**Important:** When you use multiple clearing house vendors, you run this utility for each clearing house. See "Using More Than One Payment Processor" in **BRM Configuring and Collecting Payments**.

---

Giving Refunds for Externally Initiated Payments

To make refunds for externally initiated payments, you first create the refund items. You then make the refund payments by check or other externally initiated payment method.

Create the payments outside the BRM system and then record them in the BRM database by using Payment Tool by using a refund batch. See **BRM Configuring and Collecting Payments**.

**Tip:** You can create a custom application that finds refund items and sends the amount and account identification to a check-writing program.
Specifying the Minimum Amount to Refund

You can specify the minimum amount to give as a refund. The `pin_refund` billing utility processes refund items with an amount greater than the minimum you specify.

The minimum value is expressed in terms of the account currency. By default, the minimum amount is 2.

1. Open the CM configuration file (`BRM_home/sys/cm/pin.conf`) in a text editor.
2. Change the value of the `minimum_refund` entry. For example, to process refund items only for amounts greater than 3:
   ```
   - fm_pymt_pol minimum_refund 3
   ```
3. Save the file.

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

Defining Nonrefundable Items

By default, only refund items are nonrefundable. To make other credit items nonrefundable, run the `pin_bus_params` utility to change the `NonrefundableCreditItems` business parameter. For information about this utility, see the `BRM Developer’s Guide`.

To add nonrefundable item types:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```
   pin_bus_params -r BusParamsAR bus_params_AR.xml
   ```
3. In the XML file, add the nonrefundable `/item` objects after the `/item/refund` entry, separated by commas:
   ```
   <NonrefundableCreditItems>/item/refund</NonrefundableCreditItems>
   ```
   ```
   Important: Do not remove `/item/refund`.
   ```
4. Save the file as `bus_params_AR.xml`.
5. Load the XML file into the BRM database:
   ```
   pin_bus_params bus_params_AR.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 the `BRM System Administrator’s Guide`.

Reversing Refunds

If a refund fails, you can reverse the refund. Refunds can fail because of returned checks and invalid credit cards.

Note the following restrictions:

- You can reverse only failed refunds.
A refund can be reversed only once. You cannot reverse a refund that has been reversed before.

To reverse a refund, write a custom application that uses PCM_OP_AR_REVERSE_REFUND. See BRM Opcode Guide.
Configuring Write-Offs

This document provides an overview of the Oracle Communications Billing and Revenue Management (BRM) write-offs and the write-off reversal feature.

See also:
- List of Accounts Receivable Features
- Opening and Resolving Disputes

About Write-Offs

**Important:**
- Before you write off an account or bill unit (/billinfo object), ensure that all items have been billed and that no pending or unbilled items for the account or bill unit remain.
- To ensure that you can reverse a write-off on an account, the account status must be **inactive** before you write off the account.

A write-off removes from your company’s assets an A/R amount that your company has determined the customer will never pay. A write-off can also remove an A/R amount that your company has decided it will not refund to the customer; for example, if the amount is very small or if you sent the customer a check that was returned because the customer moved without leaving a new address.

For write-offs, you *always* write off the entire account, bill, or bill item.

You can write off the following:
- **Accounts**: Write-offs are not available for an account that owns a top-level paying bill unit if its Due is zero. The account should be inactivated before it is written off.
- **Bills**: Write-offs are not available for a bill if it is pending or its Due is zero.
- **Bill items**: Item level write-offs are allowed for pending and open items.

The following types of events are created for a write-off:
- A write-off event holding the total net amount of the write-off with a G/L ID for the net amount
- A tax write-off event holding the VAT amount of the write-off with a G/L ID for tax amount

To perform a write off:
The items to write off must be open.

The write-off amount must be less than or equal to the amount due.

BRM creates a write-off item for the amount to be written off. It transfers the Due from the write-off item to the bill items being written off and closes the bill items.

Writing off uncollectable debt or an unpayable credit lets you control how it is treated in your accounting and G/L reporting system. Depending on how your company has set up its G/L system, the amount written off is transferred from A/R to a bad debt account or, for unpayable credit, to a miscellaneous revenue account.

A write-off always includes taxes, but you can specify whether the net and tax amounts are written off in one or two events. If you write off in one event, the net and tax amounts are stored separately within the event and can be mapped to different sets of G/L accounts. If you write off in two events, one event stores the net amount of the write-off while another event stores the tax amount. The net and tax amounts stored in these two events can also be mapped to separate sets of G/L accounts. However, the way you specify mapping information for G/L accounts is different for the one-event and the two-event write-off. For more information on how you map the net and tax amounts of a write-off to separate sets of G/L accounts, see "Assigning G/L IDs to Nonrated Events."

You can use the PCM_OP_BILL_POL_VALID_WRITEOFF to customize how items are validated for a write-off. See BRM Opcode Guide.
Configuring Write-Off Reversals

This document provides an overview of the Oracle Communications Billing and Revenue Management (BRM) write-off reversal feature.

Topics in this document:

- About Write-Off Reversals
- About Overpayment and Underpayment Allocation to a Written-off Account or Bill
- Defining Reason Codes for Write-Off Reversals
- Mapping G/L IDs to Write-Off Reversal Events
- Enabling Automatic Write-Off Reversals during Payment Collection
- Enabling Write-Off Reversals for Bill Units

See also:

- List of Accounts Receivable Features
- Configuring Write-Offs

About Write-Off Reversals

Write-off reversals enable you to re-allocate payments to accounts and bill units that were written off due to unpaid balances. Unpaid bills and item charges are generally written off as unrecoverable debt, but this feature enables the amount written off to be recovered and allocated to open bills and items of an account.

The following steps occur when a payment is made for a written-off amount. It also describes the steps that occur if the original payment must be reversed and the paid amount must be written off again:

1. An account or bill unit contains an unpaid balance that was written off as unrecoverable, and a payment is received for the written-off amount.

2. If the write-off reversal feature is enabled, the write-off on the account or bill unit is reversed.

3. The written-off balance is transferred back to the account or bill unit.

4. The payment is applied to the account or bill unit and all paid-off bill items are closed.

5. Any remaining balance on the account or bill unit (for an underpayment) is written off again.

6. If the payment fails, a reversal of the payment is submitted.
7. BRM verifies that the original payment was applied to the written-off account or bill unit.

8. The second write-off on the account or bill unit is reversed, if one exists (when the write-off reversal payment was an underpayment).

9. Any bill items closed by the original payment are reopened and the amount is transferred back to the account or bill unit.

10. The original payment is reversed.

11. After the payment reversal, if the account or bill unit has a balance due, the amount is written off again to restore the account balance to 0, and the account or bill unit status is reset to write-off.

**About Overpayment and Underpayment Allocation to a Written-off Account or Bill**

After a payment is posted to a written-off account or bill unit, the handling of the bill or bill items is determined by the amount of the payment: exact payment, overpayment, or underpayment. Exact payments are allocated to the appropriate bill and bill items, which are closed after being paid, and the account’s write-off flag is not reset to write-off.

Overpayments and underpayments are allocated according to the rules defined in your business policies:

- In general, if the payment amount is **more than** the amount written off, the paid bills and bill items are closed and the remaining amount is left unallocated in the account. The account or bill unit is left open and the account’s write-off flag is not reset to write-off.

- If the payment amount is **less than** the amount written off, the payment is fully allocated, and the open bills and bill items are closed again. The remaining amount is written off and the account’s write-off flag is reset to write-off.

- If an underpayment fails and you reverse it, both the amount that was written off (due to the underpayment) and the amount that was removed by the payment are returned to the account or bill unit. This restores the account balance to its original state so the entire amount can be written off again. This corresponds to the initial write-off amount for the account or bill unit.

For example, if your company writes off a $50 unpaid balance on an account or bill unit and 6 months later receives a payment of $45 for that account or bill unit, the $50 write-off is returned to the account or bill unit so that the payment can be applied toward the balance. After the payment amount is subtracted, the remaining $5 balance is written off again.

Later, if the $45 payment fails, the following steps occur when you reverse the payment:

1. The $5 write-off amount is returned to the account balance.

2. The $45 balance that was removed by the initial payment is returned to the account balance.

3. The $50 account balance is written off again.

- If any open unallocated items are in the account at the time of the reversal, the re-write-off on the account does not occur. You must first allocate and close the
open items, or customize the PCM_OP_BILL_POL_REVERSE_PAYMENT policy opcode to allocate and close the open items, before performing the reversal.

Defining Reason Codes for Write-Off Reversals

You define additional reason codes in the reasons.locale file for the write-off reversal event and the re–write-off of an account. You use the Credit Reasons domain (version 8) for write-off reversal reasons, and you should map the reason code to the /event/billing/writeoff G/L ID, as shown in the following sample:

```
DOMAIN = "Reason Codes-Credit Reasons" ;
STR
  ID = 4 ;
  VERSION = 8 ;
  STRING = "Write-off for Auto-writeoff reversal feature" ;
  EVENT-GLID
    "/event/billing/writeoff" 110;
  EVENT-GLID-END
END
```

**Note:** If you add your own reason codes to the reasons.locale file, you should use IDs above 100,000.

To define reason codes for write-off reversals, you edit the reasons.en_US sample file in the BRM_home/sysmsgs/reasoncodes directory. You then use the load_localized_strings utility (see "load_localized_strings" in BRM Developer’s Guide) to load the contents of the file into the /strings objects.

When you run the load_localized_strings utility, use the following command:

```
load_localized_strings reasons.locale
```

**Note:** If you are loading a localized version of this file, use the correct file extension for your locale. For a list of file extensions, see "Locale Names" in BRM Developer’s Guide.

For information on loading the reasons.locale file, see "Loading Localized or Customized Strings" in BRM Developer’s Guide. For information on creating new strings for this file, see "Creating New Strings and Customizing Existing Strings" in BRM Developer’s Guide.

Mapping G/L IDs to Write-Off Reversal Events

You can track funds you recover from write-off reversals in your general ledger (G/L) system. In addition, you can track the following events that occur if the payment fails and must be reversed:

- If the payment was an underpayment, the write-off reversal event that occurred for the second write-off, which was required to write off the unpaid balance on the account.
- The reversal of the original payment that failed.
- The re–write-off of the debt on the account.
Enabling Automatic Write-Off Reversals during Payment Collection

Depending on your G/L configuration, you can create single or separate G/L entries to track the reversed written-off amount and associated tax amount in respective G/L accounts.

You map custom G/L IDs for write-off reversal events by editing and loading the `reasons.locale` file. This file contains the reason code definition that is assigned to the G/L ID of the `/event/billing/writeoff` event.

To load a customized `reasons.locale` file into the BRM database, use the `load_localized_strings` utility. This utility loads the event-to-G/L ID mapping into a `/config/map_glid` object in BRM. The G/L ID-to-event mapping is defined in the `pin_glid` file.

You map custom G/L IDs when you define your reason codes for the write-off reversal event. See "Defining Reason Codes for Write-Off Reversals".

For more information about general ledger data, see "About Collecting General Ledger Data" in BRM Collecting General Ledger Data.

Enabling Automatic Write-Off Reversals during Payment Collection

You can configure BRM to reverse write-offs on accounts and bill units automatically when payments are received for written-off amounts. This enables the payment to be posted to the accounts and bill units and the written-off amount to be recovered immediately during payment processing.

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

To enable automatic write-off reversals during payment collection:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```
   pin_bus_params -r BusParamsAR bus_params_AR.xml
   ```
3. In the XML file, change `disabled` to `enabled`:
   ```
   <AutoWriteOffReversal>enabled</AutoWriteOffReversal>
   ```
4. Save the file as `bus_params_AR.xml`.
5. Load the XML file into the BRM database:
   ```
   pin_bus_params bus_params_AR.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.

For information on how to customize the write-off level to search for and to be available for reversal during payment processing, see BRM Opcode Guide.

Enabling Write-Off Reversals for Bill Units

By default, BRM is configured to reverse write-offs for accounts when payments are received.
You can configure BRM to reverse write-offs for bill units when payments are received for a written-off bill unit. When you configure BRM to perform write-off reversals for bill units, BRM reverses write-offs for both bill units and accounts.

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

To enable write-off reversals for bill units:

1. Go to `BRM_home/sys/data/config`.
2. Create an XML file from the `/config/business_params` object:
   ```
   pin_bus_params -r BusParamsAR bus_params_AR.xml
   ```
3. In the XML file, change `account` to `billinfo`:
   ```
   <WriteOffLevel>billinfo</WriteOffLevel>
   ```
4. Save the file as `bus_params_AR.xml`.
5. Load the XML file into the BRM database:
   ```
   pin_bus_params bus_params_AR.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`.  

This document describes how to create and manage account (/group/billing object) and bill unit (/billinfo object) hierarchies in your Oracle Communications Billing and Revenue Management (BRM) system.

See also:
- List of Accounts Receivable Features
- BRM Concepts

**About Account Groups**

Accounts can be organized into groups for billing purposes or to represent the relationships between the accounts graphically in Billing Care. There are several kinds of account groups:

- **Hierarchical**: An account hierarchy has a parent account and any number of child accounts and other account hierarchies. Account hierarchies are created for the following reasons:
  - Solely to display relationships between accounts. In such cases, the parent account pays *none* of a child account’s charges.
  - To provide a mechanism that custom reports can use to identify related accounts and analyze those relationships.
  - To provide the organizational structure for rolling charges of child accounts up to the parent account during billing.

---

**Note:** Because bill unit hierarchies do not have to match account hierarchies, you do not need to set up account hierarchies to roll up charges among accounts. Typically, however, bill unit hierarchies exist within the structure of an account hierarchy, and applications such as Billing Care facilitate the creation of paying and nonpaying bill unit relationships during account hierarchy setup.

For more information, see "About Bill Unit Hierarchies."

---

A child account can belong to only one parent account.

See "Creating and Managing Account and Bill Unit Hierarchies."
About Account Groups

- **Charge or Discount Sharing:** A charge or discount sharing group consists of an owner account or service and one or more member accounts or services. Charge or discount sharing groups are created so that charges and discounts can be shared among accounts.

  See "Creating and Managing Charge and Discount Sharing Groups".

- **Sponsored top-up:** A sponsored top-up group consists of an owner account and one or more member accounts. The owner account can top up a specified balance in each member account. See "About Topping Up Accounts" in BRM Configuring and Collecting Payments.

  In all types of account groups, the bills and payments belong to the account’s bill units. See "About Bill Units and Account Groups".

### About Bill Units and Account Groups

Accounts can have one or more bill units. Each bill unit stores billing information and tracks the charges for a particular bill. When accounts are set up with group relationships, the payment type of the bill units in the accounts, not the accounts themselves, determine which accounts pay the charges.

In account groups, bill units have an additional internal group structure:

- In an account hierarchy, the bill unit hierarchy can, but does not have to, match the account hierarchy. If necessary, a bill unit hierarchy can span multiple account hierarchies (that is, it can contain bill units of accounts that belong to different account hierarchies).

  Typically, however, an account hierarchy has a top-level parent bill unit and any number of child bill units and other bill unit hierarchies. The charges of the hierarchy’s nonpaying child bill units are rolled up to the top-level paying parent bill unit during billing.

  If an account hierarchy is set up solely to display relationships among accounts, the bill unit in the parent account pays none of the charges in a child account’s bill unit. Both the parent and child bill units are paying bill units.

  A child bill unit can belong to only one parent bill unit.

  See "Creating and Managing Account and Bill Unit Hierarchies".

- A discount sharing group or charge sharing group consists of a group owner and one or more members. The group owner’s account has an owning balance group. The owning balance group bears the financial impact of the sharing group, and the bill unit for this balance group pays for the member bill unit’s charges.

  Member accounts can have more than one bill unit. If a member account has multiple bill units, the bill unit that benefits from sharing is the one that has the balance group whose service has been chosen for sharing.

  See "Creating and Managing Charge and Discount Sharing Groups".

### Comparing Hierarchy and Sharing

This section describes the main differences between account and bill unit hierarchies and charge and discount sharing groups.

#### Relationships

- **Hierarchy:** Creates parent-child relationships among accounts or bill units.
• **Sharing**: Creates owner/member relationships in which the owner assumes certain charges for the member or shares balances with the members. Discount sharing is based on discount offers purchased by the owner as part of charge offer charges. Charge sharing is based on chargeshare offers.

**Bill Units and Balance Groups**

• **Hierarchy**: Parent and child accounts have either paying bill units or nonpaying bill units.

• **Sharing**: Charge and discount sharing group owner accounts have owning balance groups that are part of bill units. For charge sharing, charges for eligible member events impact the owning balance group and associated bill unit. For discount sharing, shared discounts impact the member’s balance groups by either increasing noncurrency balances or reducing the currency impact of an event.

**Charge Offer and Service Ownership**

• **Hierarchy**: Nonpaying child bill units of the same parent bill unit do not have to own the same charge offers.

• **Sharing**: An owner account does not need to have the same charge offers or services as the members.

The owner bill units can provide sharing to group members even if, at group creation time, some members do not own the services for which sharing is provided. In this case, only members who currently own the shared services benefit from charge or discount sharing. Members who do not own the service when sharing is set up can participate in sharing if they purchase that service in the future.

**Charge Offer Guidelines**

• **Hierarchy**: A parent bill unit can pay for any charge offer.

• **Sharing**: An owner can pay for a member’s charges in accordance with a chargeshare offer selected when the charge sharing group is created. An owner can share any discount included in the packages they purchase.

**Number of Parents or Owners**

• **Hierarchy**: A child account can belong to only one parent account, and a child bill unit can belong to only one parent bill unit.

• **Sharing**: Multiple owners can provide charge or discount sharing for the same service, and a member account can participate in multiple sharing groups for a service. Thus, a member bill unit can share charges with multiple owner bill units or receive discounts from multiple owners.

**What the Parent or Owner Pays For**

• **Hierarchy**: A parent bill unit pays all the charges for its nonpaying child bill units. The parent bill unit cannot pay for only some of a nonpaying child bill unit’s charges.

Bill items are created for nonpaying bill units, but the amounts in them are rolled up at billing to the paying bill unit’s bill.

• **Sharing**: An owner bill unit pays only the member charges that are defined as shared when the charge sharing group is created.

A charge sharing group can include some or all chargeshare offers in the database. The charges go directly to the bill items of the sharing group owner’s bill unit.
Multiple Bill Units and Groups

- **Hierarchy:** If a child account has multiple bill units, the parent account can pay the charges for none, one, several, or all of the child account’s bill units, depending on whether the bill units are paying or nonpaying.

- **Sharing:** If a member account has a service that participates in multiple charge and discount sharing groups from different owner accounts, the member’s charges are distributed among the groups’ owners and the member benefits from each owner’s shared discounts. BRM distributes the charges among the owner bill units and applies discounts based on priorities defined when the member joins the sharing groups. The member account pays any charges that are not eligible for sharing.

Collecting Payment Information

- **Hierarchy:** During account creation, payment information is not collected for child accounts’ nonpaying bill units. Their payment method is **Paid by Parent Account**.

- **Sharing:** During charge sharing group creation, the payment method of the owning balance group is set as the payment method for any shared charges. The members’ payment method has no bearing on the sharing, so members can use any payment method.

  Discounts do not generate bills, they only change the amount of a bill. Therefore, payment method is not a factor for discount sharing groups.

Tracking Balance Impacts

- **Hierarchy:** Nonpaying bill units track and display their own balance impacts in real time. When you bill accounts in a hierarchy, balance impacts of nonpaying bill units are rolled up to their paying parent bill unit, and the account that owns the paying parent bill unit is billed for them. The account that owns the nonpaying bill unit and the account that owns the paying bill unit can be in different account hierarchies.

- **Sharing:** Member bill units do not track or display the balance impacts of shared charges. The owning balance group tracks the balance impacts of usage resulting from noncurrency shared discounts; for example, included minutes.

Balances

- **Hierarchy** Account hierarchies do not affect any type of balance. Bill unit hierarchies are concerned only with billing, so only currency balances are affected.

- **Sharing:** Charge and discount sharing groups affect all types of balances. For example, you can apply an owner’s discount for included minutes to each member or apply a 15% discount on a member’s monthly fee.

Hierarchy Balance Impacts

The parent-child relationship has no financial impact unless a child bill unit is a nonpaying bill unit. Nonpaying bill units maintain all their charges until billing, when the charges are rolled up to their paying parent bill unit, which is then responsible for the bill.

No financial impact occurs when you move an independent account or bill unit into a hierarchy and make it a child. If you make the bill unit nonpaying, you choose whether to bill it for bill-in-progress charges or to transfer the charges to the new paying parent bill unit.
When you move a nonpaying bill unit from one paying parent bill unit to another, you can specify which parent pays the bill-in-progress charges. By default, the bill-in-progress charges transfer to the new parent bill unit.

Before you can move a nonpaying bill unit out of a bill unit hierarchy, you must change it to a paying bill unit and specify whether the bill unit or its former parent will pay any bill-in-progress charges.

**Sharing Group Balance Impacts**

When a member account’s service is added to a charge or discount sharing group, a financial relationship is created between the sharing group owner account and the member account.

- **Discount sharing:** If the owner account is active and the member account has services eligible for discount sharing, the shared discounts reduce the amount of money a member owes by applying the owner’s discounts to the balance of the member account before finalizing the member’s charges.

- **Charge sharing:** If the owner account is active and the member service has events eligible for charge sharing, the shared charges impact the balance of the owner account rather than the member account.

The member account stops benefiting from charge and discount sharing groups in the following situations:

- The owner account is inactive or closed.
- The sharing group is deleted from the owner account.
- The member account’s service is removed from the sharing group.
- Events generated by a member fall outside of the validity period for the shared charge or discount.
- The charges or discounts that were included in the sharing group are deleted through PDC.
- For discount sharing groups, the noncurrency balances that were offered as part of the discount have been depleted in the owner account (for example, a monthly discount of included minutes that has already been consumed by the owner or other members). In this case, the member still benefits from currency-type shared discounts, such as a 10% reduction on bills for email usage or a $15 discount on overseas calls.

**About Account Hierarchies**

An account hierarchy is a set of accounts organized according to their positions in relation to each other. The relationships among accounts in a hierarchy are similar to parent-child relationships. The hierarchy is headed by a parent account with child accounts beneath it. At each level above the bottom of the hierarchy, the child accounts themselves can be parent accounts.

You set up account hierarchies for the following reasons:

- Solely to display relationships among accounts. In such cases, the parent account pays none of a child account’s charges.
- To provide a mechanism that custom reports can use to identify related accounts and analyze those relationships.
To provide the organizational structure for rolling charges of child accounts up to the parent account during billing.

Note:  Because bill unit hierarchies do not have to match account hierarchies, you do not need to set up account hierarchies to roll up charges among accounts. Typically, however, bill unit hierarchies exist within the structure of an account hierarchy, and applications such as Billing Care facilitate the creation of paying and nonpaying bill unit relationships during account hierarchy setup.

For more information, see "About Bill Unit Hierarchies."

Performance Impact of Account Hierarchies

To maintain data consistency, many operations lock an account at the beginning of a transaction. Therefore, in an account hierarchy, many of the associated accounts are also locked. Although this provides reliable data consistency, it can cause a lot of serialization, which decreases the throughput of the system.

If, with account hierarchies, this problem affects your system, you can choose to lock specific balance groups instead of the whole account.

Important:  Balance group locking might enable separate contexts to attempt to lock the same object, causing a system halt. When balance group locking is used, every feature that uses it should be examined for overlap.

For more information about balance group locking, see "Locking Specific Objects" in BRM Developer’s Guide.

About Bill Unit Hierarchies

Bill unit hierarchies are a set of bill units organized according to their positions in relation to each other. A bill unit hierarchy is headed by a parent bill unit with child bill units beneath it. At each level above the bottom of the hierarchy, the child bill units themselves can be parent bill units.

Typically, a bill unit hierarchy exists within the context of an account hierarchy used for billing purposes. The bill unit hierarchy can, but does not have to, match the
account hierarchy. If necessary, a bill unit hierarchy can span multiple account hierarchies (that is, it can contain bill units of accounts that belong to different account hierarchies).

A bill unit’s position in a hierarchy does not necessarily indicate whether it pays its own bills. Any bill unit, either a parent or child, can be a paying bill unit or a nonpaying bill unit, and both parent and child accounts can have paying bill units and nonpaying bill units:

- A paying bill unit pays its own bill and, if it is a parent, the bills of its nonpaying child bill units.
- A nonpaying bill unit’s bill is paid by its parent bill unit. If its parent is also a nonpaying bill unit, its charges are rolled up the bill unit hierarchy to the next paying ancestor bill unit.

Figure 9–1 shows a simple bill unit hierarchy. The hierarchy contains three bill units, one in each account. Because the paying bill unit in the parent account pays the bill for the nonpaying bill unit in the child account, however, only two bills are generated for the three accounts.

**Figure 9–1  Simple Bill Unit Hierarchy**

Account and bill unit hierarchies do not have to match. For example, a parent bill unit does not have to belong to a parent account, and a child bill unit does not have to belong to a child account.

Figure 9–3 shows an account hierarchy containing one parent account and one child account. Each account contains a bill unit. The parent account’s bill unit is the nonpaying child of the child account’s paying parent bill unit. In this situation, the child account pays the parent account’s charges.
When hierarchical accounts have multiple bill units, the bill unit hierarchy becomes more complex. A child account can have both nonpaying bill units and paying bill units. The parent account is not required to pay all of the child account’s bills.

Figure 9–3 shows a parent account and a child account with two bill units each. One of the child account's bill units is a paying bill unit. The charges for that bill unit are paid by the child account, not by the parent account. In this case, three bills are generated: one for the parent account, one for the parent-child account, and one for the child account.

At each level above the bottom of a hierarchy, the child bill units themselves can be parent bill units.

Figure 9–4 shows a three-level account hierarchy containing seven bill units. Because only three of the seven bill units are paying, only three bills are generated. The top parent account receives two bills, and the bottom child account receives one bill.
A bill unit hierarchy can contain bill units of accounts that belong to different account hierarchies.

Figure 9–5 shows two account hierarchies with a total of five bill units. The bill units are grouped into two bill unit hierarchies, each containing bill units from both account hierarchies. Because only two bill units are paying, only two bills are generated. One parent account receives a bill, and one child account receives a bill.
For information about creating bill unit hierarchies, see "Creating Account and Bill Unit Hierarchies".
How Account Status Changes Affect 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 example, Figure 9–6 shows the bill units that are inactivated in a multilevel hierarchy when a parent account is inactivated.

Figure 9–6 Effect of Parent Account Status Change on Bill Unit Hierarchy

Changing the status of a paying parent bill unit changes the status of all accounts that have at least one nonpaying child bill unit of the paying bill unit.

For example, Figure 9–7 shows the bill units that are inactivated in a multilevel hierarchy when the status of an individual paying bill unit is inactivated.
Calculating Balance Due in Account and Bill Unit Hierarchies

**Currency Requirements of Hierarchies**

Nonpaying bill units must have the same currency as the account that owns their parent bill unit. If the accounts that own the parent and child bill units use two currencies, their primary and secondary currencies must match.

**Billing Setups in Hierarchies**

Because paying bill units handle the billing for nonpaying bill units, nonpaying bill units must have the same billing day of month (DOM), billing frequency, accounting cycle, and language as their paying parent bill unit.

**Calculating Balance Due in Account and Bill Unit Hierarchies**

An account hierarchy is a set of accounts organized according to their positions in relation to one another. Each hierarchy consists of one parent account and any number of child accounts and other account hierarchies.

When the accounts in a hierarchy are billed, however, the relationships among the accounts’ bill units, not among the accounts, determine whether an account pays its own charges or rolls them up to another account inside or outside the hierarchy. To keep the charges within the account hierarchy, each bill unit in the hierarchy’s accounts should be part of a bill unit hierarchy that is contained within the account hierarchy.

---

**Note:** Bill unit hierarchies can, but do not have to, match account hierarchies. See About Bill Unit Hierarchies.

---

Child accounts can have a paying bill unit or a nonpaying bill unit. The balance due is billed differently for accounts that have paying bill units and nonpaying bill units:

- The balance due for a paying bill unit is billed to itself.
- The balance due for a nonpaying bill unit is billed to its paying bill unit (the nonpaying bill unit’s first paying bill unit ancestor).
Calculating Balance Due in Account and Bill Unit Hierarchies

BRM creates bills and bill items for all account bill units: parent, child, paying, and nonpaying. Billing, however, involves two different processes:

- Changing the status of bills and bill items to open, and creating new pending bills and bill items for the next bill. This occurs for all accounts.

  **Note:** If a nonpaying child bill unit is billed before its paying parent bill unit, the nonpaying bill unit's items remain pending until the paying bill unit has been billed.

- Requesting a payment for the bill (for example, initiating a credit card transaction). This occurs for paying bill units only. A nonpaying bill unit never receives a payment request.

Each bill unit includes a pending bill and one or more pending bill items. As an account incurs balance impacts, such as usage fees, the balance due accumulates in the pending bill items.

Each bill item includes these fields:

- **PIN_FLD_ACCOUNT_OBJ:** The account that the item belongs to. This field always points to the account that owns the item.
- **PIN_FLD_BAL_GRP_OBJ:** The balance group that the item belongs to.
- **PIN_FLD_BILLINFO_OBJ:** The bill unit that the item belongs to.
- **PIN_FLD_BILL_OBJ:** The bill that the item belongs to. This field always points to the account’s own bill.
- **PIN_FLD_AR_BILLINFO_OBJ:** The paying bill unit that the item belongs to.
- **PIN_FLD_AR_BILL_OBJ:** The paying bill that the item belongs to.

The AR_BILLINFO_OBJ and AR_BILL_OBJ fields determine which account, bill unit, and bill handles billing for the item. If an item belongs to a nonpaying bill unit, the item’s AR_BILLINFO_OBJ field points to the nonpaying bill unit’s paying bill unit, and the AR_BILL_OBJ field points to a bill in the parent account of the paying bill unit.

In a bill unit hierarchy with more than two levels, the balance due belongs to a nonpaying bill unit’s first paying bill unit ancestor and its corresponding bill.

Figure 9–8 shows a parent account, a child account with a nonpaying bill unit, and a child account with a paying bill unit. Notice that in the child account with the nonpaying bill unit, the paying bill field (AR_BILL_OBJ) and paying bill unit field (AR_BILLINFO_OBJ) point to the parent account.
Who Pays for Open Items and Pending Items?

An account might have open items and pending items. This might happen if the customer has not paid an open bill or has paid only part of it. In that case, there would be an open item and a pending item.

If that account’s bill unit changes from nonpaying to paying or from paying to nonpaying, the payment responsibility for only pending items is affected. Payment responsibility for open items is not changed.

Figure 9–9 shows a child account with a nonpaying bill unit that was a parent account with a paying bill unit, so it has a pending item and an open item. The open item points to the child account as the owner of the paying bill unit (AR_BILLINFO_OBJ), but the pending item points to the parent account as the owner of the paying bill unit.
In some cases, an account can accumulate balance impacts for part of a billing cycle before its bill unit becomes nonpaying. When the bill unit becomes a nonpaying child, the paying parent bill unit becomes responsible for pending charges accumulated before the child bill unit became nonpaying.

Figure 9–10 shows an account that is billed on the 5th day of each month. The account’s bill unit becomes a nonpaying child on the 10th day of the month, but because it has already incurred balance impacts recorded in a pending item, the paying parent bill unit is billed for the balance due accumulated from the 5th through the 15th.

Figure 9–10  Account Charges before Bill Unit Became Nonpaying

![Diagram showing account charges before bill unit became nonpaying](Image)

Table 9–1 summarizes changes to payment responsibilities:

<table>
<thead>
<tr>
<th>Change to Account</th>
<th>Open Items</th>
<th>Pending Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent account with paying bill unit becomes child account with nonpaying bill unit.</td>
<td>Nonpaying bill unit in the child account is responsible for its open item balance due.</td>
<td>Paying bill unit in the parent account is responsible for the pending item balance due of the child account’s nonpaying bill unit.</td>
</tr>
<tr>
<td>Nonpaying bill unit in a child account becomes a paying bill unit.</td>
<td>Paying bill unit in the parent account is responsible for the open item balance due of the former child account’s nonpaying bill unit.</td>
<td>Paying bill unit in the child account is responsible for its pending item balance due.</td>
</tr>
<tr>
<td>Child account with nonpaying bill unit changes parent account and paying parent bill unit.</td>
<td>Paying bill unit in the old parent account is responsible for the open item balance due of the child account’s nonpaying bill unit.</td>
<td>Paying bill unit in the new parent account is responsible for the pending item balance due of the child account’s nonpaying bill unit.</td>
</tr>
</tbody>
</table>

To close all of an account’s open items before you make the account a nonpaying child, use the Bill Now feature in Billing Care.

Multiple Levels of Parent Accounts

If all child accounts (some of which might also be parent accounts) in an account hierarchy have only nonpaying bill units, the balance due for pending items is always handled by the hierarchy’s top parent account.

In Figure 9–11, account 300, which has a nonpaying bill unit, is a child of account 200, which is in turn a child of account 100. The balance due for account 300 is handled by account 100, not by account 200.
Hierarchy Changes and Billing Dates

When you make an account a child account, BRM changes the child account’s billing day of month (DOM) to match the parent account’s billing DOM.

If the child account has a nonpaying bill unit whose paying parent bill unit is in the new parent account, the nonpaying bill unit must have the same billing DOM as the paying parent bill unit.

**Important:** All bill units in a bill unit hierarchy must have the same billing DOM. BRM, however, does not enforce that requirement. Therefore, if the billing DOMs do not match in a bill unit hierarchy, you must manually align them.

The next billing date for the nonpaying bill unit, however, might not be the same as the next billing date of the paying parent bill unit. This happens because a change to a billing DOM always takes effect after the end of the current cycle.

As a result, the parent account’s paying bill unit might be billed, but the nonpaying bill unit in the child account might not be billed. Therefore, the balance due for the nonpaying bill unit in the child account is not included in the parent account bill.

For example:

1. You create Account A on June 8.
   The billing DOM is 8.
2. You create Account B on June 20 and then change its billing day to 8.
   The next billing date is August 8. This happens because the current billing cycle, from June 20 through July 20, must be completed before the billing DOM changes. A long billing cycle, from June 20 through August 8, is created.
3. You make Account B a child account of Account A, changing Account B’s bill unit to a nonpaying child of Account A’s bill unit.

4. On July 8, Account A is billed, but Account B is not because Account B’s next billing date is August 8.
   Therefore, the bill includes the balance due for only Account A, the parent account.

5. On August 8, both accounts are billed.
   The bill includes the balance due from both accounts:
   – Parent account balance due from July 8 to August 8.
   – Child account balance due from June 20 to August 8.

6. On all subsequent billing dates, both accounts are billed.

Figure 9–12 shows a time line for the billing dates for two accounts in a parent-child relationship where the child account has a nonpaying bill unit.

**Examples of Changes to Account Hierarchies**

These examples show who is responsible for the balance due following various account hierarchy changes.

**Note:** Except where noted, these examples assume that the child account’s nonpaying bill unit is owned by the parent account’s paying bill unit.

**An Account Becomes a Child Account with a Paying Bill Unit**

In this case, there is no change to how the balance due is handled. The parent account handles its own balance due; the child account handles its own balance due.
An Account Becomes a Child Account with a Nonpaying Bill Unit When It Is Created
In this case, the total balance due is handled by the parent account, which owns the paying bill unit, and the billing date of the child account is the same as the parent account.

An Account Becomes a Child Account with a Nonpaying Bill Unit Immediately after It Is Created
In this case, the account has not been billed yet, so all its items are pending. Therefore, its total balance due is handled by the parent account, which owns the paying bill unit.

---

**Note:** By default, for customers who pay by credit card, BRM charges purchase fees and the first cycle forward fee at account creation. The balance due for these fees is stored in open items and is charged to the child account’s bill unit. You can turn off credit card collection at account creation by editing the `cc_collect` entry in the Connection Manager (CM) configuration file (BRM_home/sys/cm/pin.conf). See “Charging Customers at Account Creation” in BRM Managing Customers.

---

An Account Becomes a Child Account with a Nonpaying Bill Unit Several Months after It Is Created
In this case, the balance due for all pending items in the child account’s nonpaying bill unit is handled by the parent account, which owns the paying bill unit. Any open items are handled by the child account. The billing date of the nonpaying bill unit in the child account is changed to match the billing date of the account that owns the paying bill unit.

A Child Account with a Nonpaying Bill Unit Changes Parent Accounts

**Note:** This example assumes that the nonpaying bill unit’s paying parent bill unit was also changed to a paying bill unit owned by the new parent account.

---

In this case, the balance due for all pending items in the nonpaying bill unit of the child account is now handled by a different parent account and that parent account’s paying bill unit. Any open items are the responsibility of the former parent account. The billing date of the child account is changed to match the billing date of the new parent account, which owns the current paying parent bill unit.

A Child Account with a Nonpaying Bill Unit Becomes a Child Account with a Paying Bill Unit
In this case, the fields in pending items that specify the paying bill unit and the bill both point to the paying bill unit in the child account. If the parent account’s bill unit that formerly paid the child’s charges has any open items that include amounts due from the child account, the parent account’s bill unit is responsible for those amounts even when the child account’s bill unit is no longer nonpaying.
Creating Account and Bill Unit Hierarchies

You can create account and bill unit hierarchies by using Billing Care or by using custom applications to call BRM opcodes.

By default, accounts are created with one bill unit. When you create account hierarchies in Billing Care, the bill units are automatically assigned the same hierarchical position as the accounts to which they belong.

To enable customers to receive separate bills for different services, you can create additional bill units per account. You then specify whether the bill units are paying or nonpaying and parents or children.

The bill units of accounts that do not belong to the same account hierarchy can form a bill unit hierarchy. Nonpaying child bill units in one account hierarchy can have a paying parent bill unit in a different account hierarchy.

Managing Account Hierarchies

To manage account hierarchies, you can do the following:

- Display account hierarchies in Billing Care.
  You can see the structure of an account hierarchy in the Hierarchy tab of the hierarchy’s parent account. Initially, this tab shows the direct lineage of an account, not its siblings or the siblings of its parent.

- Change the parent of an account.
  You can use Billing Care to change an account’s parent at any time. You change an account’s parent by adding the account to a hierarchy, removing it from a hierarchy, or moving it from one hierarchy to another.

- Defer account hierarchy changes until a later date.
  You can schedule a parent change for a future date. You can then use a daily billing utility, pin_deferred_act, to execute the change automatically on the scheduled date.

  "Managing Deferred Actions" in BRM Managing Customers.

Moving Closed Accounts into or out of Hierarchies

By default, BRM does not enable you to move closed accounts into or out of account hierarchies. To enable that functionality, configure the allow_move_close_acct entry in the CM configuration file.

1. Open the CM configuration file (BRM_home/sys/cm/pin.conf) in a text editor.

2. Set the allow_move_close_acct entry to 1.

   By default, this entry is set to 0.
   - fm_bill allow_move_close_acct 1

3. Save and close the file.
This document describes how to create and manage charge and discount sharing groups in your Oracle Communications Billing and Revenue Management (BRM) system.

See also:
- Creating and Managing Account and Bill Unit Hierarchies
- List of Accounts Receivable Features

**About Charge and Discount Sharing Groups**

Accounts or services can share charges or discounts by joining groups that consist of an owner account or service and one or more member services:

- **Discount sharing group.** In a discount sharing group, the owner shares its discounts with the members, as shown in Figure 10–1:

  ![Discount Sharing Group Diagram](image)

  For more information, see "About Discount Sharing Groups".

- **Charge sharing group.** In a charge sharing group, the owner assumes charges that are incurred by the members, as shown in Figure 10–2:
About Charge and Discount Sharing Groups

10-2
Managing Accounts Receivable

Figure 10–2 Charge Sharing Group

Charge sharing group

Group owner

Member charges

Group member 1
Group member 2
Group member 3

For more information, see "About Charge Sharing Groups".

You can use Billing Care to set up charge and discount sharing groups, or you can customize a third-party client application to set up sharing. For information on customizing charge and discount sharing, see BRM Opcode Guide.

Working with Complex Charge and Discount Sharing Groups

Networks of charge and discount sharing groups can be complex. An owner can have more than one charge or discount sharing group, and a member can belong to more than one charge or discount sharing group, as shown in Figure 10–3:

Figure 10–3 Complex Charge and Discount Sharing Groups

<table>
<thead>
<tr>
<th>Charge sharing group A</th>
<th>Charge sharing group B</th>
</tr>
</thead>
<tbody>
<tr>
<td>100% of GSM services</td>
<td>50% of email services</td>
</tr>
</tbody>
</table>

Owner = Mom

Member = Louise
Member = Dave

Member = Paul
Member = Anika

Discount sharing group C
10% off of GSM services

Owner = Dad
Member = Tony

Member = Jessie

In this example, Mom owns two charge sharing groups: one for GSM services and one for email services. She assumes charges for all her children: 100% of GSM charges for Louise and Dave; 50% of email charges for Paul and Anika; and both GSM and email charges for her youngest son, Tony. Dad owns a discount sharing group; he shares a 10% discount on GSM charges with his son Tony and brother Jessie. The members have the following benefits:

- When Paul and Anika use email, Mom assumes part of their charges. They have no discounts, shared or owned, so Mom pays 50% of their monthly email charges and they pay the remaining 50%.
- When Louise and Dave use GSM services, Mom assumes their charges. They have no discounts, so Mom pays 100% of their monthly GSM charges.
About Discount Sharing Groups

When Jessie uses GSM services, he receives a 10% discount from Dad. Jessie is not a member of any charge sharing groups, so he pays the remaining 90% of his GSM charges.

When Tony uses email, Mom assumes part of his charges. He does not have any email discounts, so Mom pays 50% of his monthly email charges and he pays the remaining 50%.

When Tony uses GSM services, he receives a 10% discount from Dad. This discount is applied before the charges are applied. Because Tony is a member of charge sharing group A, Mom pays the remaining 90% of his GSM charges.

In more complicated networks, an owner of a charge or discount sharing group can also be a member of another charge or discount sharing group. For example, Mom owns a charge sharing group that assumes email charges for Paul, Anika, and Tony. Mom, in turn, is a member of Grandma’s charge sharing group, which also assumes email charges for its members. In this case, Grandma pays the email charges for Mom. In addition, she inherits the email charges for Paul, Anika, and Tony as a result of Mom’s charge sharing group.

Important: When setting up charge and discount sharing groups, avoid circular relationships. See “About Creating Charge and Discount Sharing Groups”.

About Discount Sharing Groups

Discount sharing occurs when an account or service shares its discounts with other accounts’ services. The account that shares its discounts is the owner of a discount sharing group. In the owner’s account, one of the balance groups serves as the owning balance group. The owning balance group is determined based on whether the discount sharing group owner is the account or a service in the account:

- If the group owner is the account, the owning balance group is the account’s default balance group, and all discounts purchased by the account are shared.
- If the group owner is a service in the account, the owning balance group is the associated service balance group, and only the discounts for that service are shared.

When a discount sharing group is created, a discount sharing group object (/group/sharing/discounts) is added to the database. This object contains the owner account or service, the list of discounts that are shared by the owner, and the list of members. The members are the services that use the shared discounts.

When the discount sharing group owner is a service, its service type must match the type of service to which the shared discounts apply.

When events are generated by member activity, discounts belonging to the sharing group owner are typically applied first, followed by any discounts belonging to the member account. After discounting, any remaining balance impact is applied to the member account unless the account also participates in charge sharing.

Note: You can change the order in which discounts are consumed so that discounts belonging to the member are used before shared discounts.
For members to receive discounts from the owner, the member’s ordered balance group must include a reference to the discount sharing group. This reference is created when the member joins the group. See "How Charges and Discounts Are Applied".

To create a discount sharing group, see "About Creating Charge and Discount Sharing Groups".

**How Account Status Changes Affect Discount Sharing Groups**

When the status of a discount sharing group’s owner account is changed to inactive or closed, members stop benefiting from the shared discounts as follows:

- **Owner account is deactivated**: The status of all the group’s shared discounts is changed to inactive. Members resume benefitting from the shared discounts when the owner’s account is reactivated.

- **Owner account is closed**: All the group’s shared discounts are removed from the account and from the discount sharing group.

**Note**: If keep_cancelled_products_or_discounts is set to 1 in the Connection Manager (CM) pin.conf file, the shared discounts are not removed from the discount sharing group. Instead, their status is set to canceled. For information about deleting canceled discounts, see "Deleting Canceled Discounts" in BRM Managing Customers.

**How Group Owner Changes Affect Discount Sharing Groups**

If you use a third-party client application, you can use the PCM_OP_SUBSCRIPTION_SHARING_GROUP_SET_PARENT opcode to customize the application to enable a customer service representative (CSR) to change the owner of a discount sharing group. See BRM Opcode Guide.

When the owner of a discount sharing group changes, the discounts shared by the previous owner are deleted from the group, and discounts shared by the new owner are added.

Member-generated events, including delayed events, that occur after a group owner is changed impact the new owner’s shared balances.
Members and Discount Sharing Groups

When you set up a discount sharing group, the members of the group must be services (the service balance groups).

If the owner of a shared discount is a service, the discount sharing group member service is typically the same service type as, or a subclass of, the discount owner service. For example, if the discount sharing group owner is /service/telco, the discount sharing group member might be /service/telco/gsm. This parent and subclass relationship is the standard way you set up subscription services to offer a discount based on the total usage of all subscription services in an account group.

If the owner of a shared discount is a service, BRM requires that the owner and member services be of the same type. You can specify different service types for the owner and members by implementing the relevant subscription opcodes in your custom code (see BRM Opcode Guide). However, if the owner and member services are different types, the member service must match one of the shared discount’s permitted service types (specified in the PIN_FLD_PERMITTEDS array in the /discount object).

If the owner and a member of the discount sharing group have different service types, to apply the discount to the member service, the member’s usage event type must match the event type to which the shared discount applies.

For example, you use PDC to set up a subscription discount that applies to /service/telco usage. In the Discount Attributes dialog box, you map the discount to the monthly cycle forward event. You then create a discount sharing group in which you share the telco discount that you created. You add an account to the discount sharing group and specify the account’s /service/telephony service as the member. In this case, the subscription discount is applied to the member account only when the member account generates a monthly cycle forward event for its telephony service.

When you set up a discount sharing group, you can specify that the member service be a service type or a service instance:

- **Specify a service type as the member**: When you specify a service type (for example, GSM) as a member, the events generated by all service instances in that type are considered for discount sharing. Adding members by service type is appropriate in the following cases:
  
  - You add multiple member accounts to the group to take advantage of discounts for a service type (for example, an aunt who wants to share her GSM discounts with her nieces and nephews). In this case, each niece and nephew has a different service instance, so instead of specifying each service instance individually, you specify the member accounts and the service type. This captures all of the nieces’ and nephews’ service instances under the umbrella of the GSM service type.
  
  - You want a member account that has not yet purchased a service of that type to be automatically eligible for participation in discount sharing if the member buys the service in the future. With future sharing, even though a member becomes automatically eligible, you must track the member’s purchases and, when the service is purchased, manually intervene to join discount sharing for the new service instance.

- **Specify a service instance as the member**: When you specify a service instance (for example, an employee’s work phone) as a member, only the events generated by that instance are eligible for discounting. Do this to exclude other services of that type from discount sharing.

For example, John’s account includes work email (smith@CompanyA.com) and home email (john@internetprovider.com). John’s employer wants to share a 10%
About Discount Sharing Groups

discount for monthly fees incurred only by work email. Because John’s account has both home and work services, you designate the service instance for smith@CompanyA.com as the member to prevent discount usage by john@internetprovider.com.

Currency Requirements of Discount Sharing Groups

The discount sharing group’s owner account and member accounts must use the same currency. If the accounts use two currencies, they must use the same primary currency.

Billing for Discount Sharing Groups

A discount sharing group’s owner account and member accounts can have different accounting and billing cycles. Member-generated events that qualify for discounts always consume the available shared balances.

For example, in Figure 10–4, the owner’s accounting and billing cycles begin on the 1st of the month, and the member’s accounting and billing cycles begin on the 15th of the month. The owner receives 100 free monthly minutes that are shared.

Figure 10–4 Discount Sharing between Accounts with Different Accounting Cycles

When a member generates an event that lasts 30 minutes on January 20, it consumes the available 20 minutes from the shared balance group. The member’s account receives a balance impact for the remaining 10 minutes of the event.

When another member generates an event that lasts 15 minutes on February 20, there are no available minutes in the shared balance group to consume, and the member account receives a balance impact for the entire 15 minutes.

Configuring the Start and End Times for Discount Sharing

By default, discount sharing start and end times are based on the date of the next billing cycle. Therefore:

- When you create a discount sharing group, add a discount to an existing group, or add a member to the group, discount sharing starts at the beginning of the owner’s next billing cycle.

- When you delete a discount sharing group, delete a discount from a group, or remove a member from the group, discount sharing is effective up to the end of the owner’s billing cycle.
For example, if you create a discount sharing group on January 15 but the next billing cycle does not start until February 1, the members of the group do not receive any of the owner’s shared discounts until February 1, as shown in **Figure 10–5**:

**Figure 10–5  Effective Date of Discount Sharing Group Created Mid-Cycle**

Conversely, if you delete a discount sharing group on January 15 but the next billing cycle does not start until February 1, the members of the group receive shared discounts based on the entire month of January, not just the first 15 days.

You can configure BRM to start and end discount sharing as soon as the group is created or deleted, a discount is added or deleted, or a member is added or removed. To do so, you modify an entry in the Connection Manager (CM) `pin.conf` file.

When you configure discount sharing this way, BRM prorates the shared discounts according to where the start or end date falls within the billing cycle. For example, if you delete a discount sharing group on April 15 and the owner’s billing cycle starts on the first day of each month, the members receive 50% of the discount, equivalent to the portion of the month in which discount sharing existed.

To change the discount sharing start time:

1. Open the Connection Manager (CM) configuration file (`BRM_home/sys/cm/pin.conf`).
2. Change the discount sharing startup from the beginning of the next billing cycle to the time when the discount sharing group is created or a member is added by changing the `0` to `1` in the following line:
   
   ```
   -fm_subscription propagate_discount 1
   ```
3. Stop and restart the CM and, if necessary, your client application.

### About Charge Sharing Groups

Charge sharing enables a customer to sponsor the charges of other accounts or services in the system. For example, it enables a company to pay for all of its employees’ GSM telephony services or a parent to pay for his child’s SMS and email services.

You set up charge sharing by creating a charge sharing group, which consists of the following:

- **Charge sharing group owner**: The account or service responsible for all or a portion of the charges incurred by the charge sharing group members.
- **Charge sharing group members**: The accounts and services that the owner sponsors.
- **Chargeshare offer**: The chargeshare offer specifies how and when to apply charges. It is linked to a particular event type. Chargeshare offers determine
whether events qualify for charge sharing and apply rules to calculate charge sharing amounts.

When a member incurs charges sponsored by the owner, the charges are applied to the owner’s balance group first. Any charges that remain afterward impact the member’s balance group.

Information about charge sharing groups is stored in /group/sharing/charges objects in the BRM database.

To create a charge sharing group, see “About Creating Charge and Discount Sharing Groups”.

**About Charge Sharing Group Owners**

The account that sponsors the charges is the owner of a charge sharing group. Within the owner’s account, one of the balance groups serves as the owning balance group. The owning balance group is the one that receives the balance impact of any shared charges incurred by the members.

The owning balance group is determined based on whether the charge sharing group owner is the account or a service in the account:

- If the group owner is the account, the owning balance group is the account’s default balance group.
- If the group owner is a service in the account, the owning balance group is the associated service balance group.

**How Owner Account Status Changes Affect Charge Sharing**

By default, changing the status of a charge sharing group’s owner account changes the status of all its shared charges. When the owner account is deactivated or closed, members stop benefiting from the charge sharing group. The owner no longer assumes any of the charges incurred by the members. Any new events generated by the members are charged to the member’s balance groups only.

The way that BRM treats the charge sharing group depends on whether the owner account is deactivated or closed:

- **Owner account is deactivated**: All of the group’s sponsored charges are suspended, and no new member charges are added to the owning balance group. Member services can begin benefiting from charge sharing again as soon as the owner’s account is reactivated.

- **Owner account is closed**: All of the group’s shared charges are suspended, and no new member charges are added to the owning balance group. The shared charges are removed from the charge sharing group.

---

**Note:** If `keep_cancelled_products_or_discounts` is set to 1 in the Connection Manager (CM) `pin.conf` file, the shared charges are not removed from the charge sharing group. Instead, their status is set to canceled. For information about deleting canceled discounts, see “Specifying to Delete Canceled Discounts” in *BRM Managing Customers*.  

---
About Changing Charge Sharing Group Owners

If you use a third-party client application, you can use the PCM_OP_SUBSCRIPTION_SHARING_GROUP_SET_PARENT opcode to customize the application to enable a CSR to change the owner of a charge sharing group. See BRM Opcode Guide.

When the owner of a charge sharing group changes, the charges that were sponsored by the previous owner are deleted from the group, and charges sponsored by the new owner are added.

Member-generated events, including delayed events, that occur after a group owner is changed impact the new owner’s balances.

About Charge Sharing Group Members

When you set up a charge sharing group, the members of the group must be services (the service balance groups).

If the owner of a charge sharing group is a service, the group member service is typically the same service type as, or a subclass of, the charge sharing owner service. If the owner and a member of the charge sharing group have different service types, to share charges, the member’s usage event type must match the event type to which the chargeshare offer applies.

Note: BRM requires that the owner and member services be of the same type. You can specify different service types for the owner and members by implementing the relevant subscription opcodes in your custom code. See BRM Opcode Guide.

When you set up a charge sharing group and specify the member services, you can:

■ **Specify a service type for a group of accounts as members:** When you specify multiple account instances and a service type (for example, Account 1, Account 2, and GSM) as members, only the service instances for each account are considered members. Adding members by service type is appropriate when:

  - You are adding multiple member accounts to the group and you want them all to take advantage of charge sharing for a service type (for example, an aunt who wants to pay the GSM charges for all of her nieces and nephews). In this case, each niece and nephew has a different service instance, so instead of specifying each service instance individually, you specify the member accounts and the service type. This captures all of the nieces’ and nephews’ service instances under the umbrella of the GSM service type.

  - You want a member account that has not yet purchased a service of that type to be automatically eligible for participation in charge sharing if the member buys the service in the future. With future sharing, even though a member becomes automatically eligible, you must track the member’s purchases and, when the service is purchased, manually intervene to join charge sharing for the new service instance.

■ **Specify a service instance as the member:** When you specify a service instance (for example, an employee’s work phone) as a member, only the events generated by that instance are eligible for charge sharing. Specify membership by using a specific service instance to exclude other services of that type from charge sharing.

For example, John’s account includes work email (smith@CompanyA.com) and home email (john@internetprovider.com). John’s employer wants to pay for monthly fees incurred by work email only. Because John’s account has both home
and work services, you designate the service instance for smith@CompanyA.com as the member to prevent the employer from paying for usage charges from john@internetprovider.com.

- **Specify all accounts in the system as members**: When you specify a type-only account POID as the member, events generated by any account in your system are eligible for charge sharing.

  See "About Global Charge Sharing Groups".

- **Specify all services of a specific type as members**: When you specify a type-only service POID as the member, events generated for that service type are eligible for charge sharing.

  See "About Global Charge Sharing Groups".

For member charges to be applied to the owner’s account, the member’s ordered balance group must include a reference to the charge sharing group. This reference is created when the member joins the group. For more information, see "How Charges and Discounts Are Applied".

---

**Note:** This requirement does not apply to global charge sharing groups. See "About Global Charge Sharing Groups".

---

**Currency Requirements for a Charge Sharing Group**

The charge sharing group’s owner account and member accounts must use the same currency or primary currency.

---

**Note:** The currency requirement does not apply for global charge sharing groups. See "About Global Charge Sharing Groups".

---

**Billing for a Charge Sharing Group**

A charge sharing group’s owner and member accounts can have different accounting and billing cycles. Sponsored charges are always applied to the owner account, regardless of when the member-generated events occur.

In the example illustrated in Figure 10–6, the owner’s accounting and billing cycles begin on the 1st of the month, and the member’s accounting and billing cycles begin on the 15th. Charge sharing starts on January 10 and ends on February 20, so between those dates, the owner sponsors 50% of all charges incurred by a member.

---

**Figure 10–6 Effective Dates of Charge Sharing Groups Created Mid-Cycle**

[Diagram showing effective dates of charge sharing groups created mid-cycle]
If the member generates an event on January 20 for 30 minutes at $0.10 per minute (total cost $3.00):

- The owner’s account receives a balance impact of $1.50, and the amount appears on the owner’s bill on February 1.
- The member’s account receives a balance impact of $1.50, and the amount appears on the member’s bill on February 15.

The member receives the balance impacts for all charges incurred after February 20.

**About Global Charge Sharing Groups**

You can create a charge sharing group that includes all accounts in your system or all services of a specific type, such as GSM telephony. This type of charge sharing group is called a global charge sharing group. This enables one account to pay for all or a portion of the charges incurred by anyone in your system.

You specify which events the group owner pays for and how to split charges between the owner and members by creating a chargeshare offer.

For example, you can configure BRM to charge a company for all or a portion of any call to a toll free number or any text message sent to a specified number.

Global charge sharing groups differ from charge sharing groups in these ways:

- Global charge sharing groups are not listed in a member’s ordered balance group. Therefore, the order in which charges are applied to global charge sharing groups is set by the system.
- Members in the global charge sharing group are indicated by a type-only POID.

**About the Order in Which Charges Are Applied to Groups**

The order in which BRM applies charges to the different types of charge and discount sharing groups is built into the system. BRM automatically applies charges to global charge sharing groups after applying any discounts from discount sharing groups but before applying charges to any charge sharing groups. The order in which BRM applies charges and discounts is shown below:

1. Discount sharing groups as listed in the member’s ordered balance group
2. Global charge sharing groups having type-only services
3. Global charge sharing groups having type-only accounts
4. Charge sharing groups as listed in the member’s ordered balance group

**Note:** Only charge sharing groups and discount sharing groups are listed in a member’s ordered balance group. Global charge sharing groups are not included in the list.

**About Creating Global Charge Sharing Groups**

Global charge sharing groups, like charge sharing groups, consist of an owner, members, and the chargeshare offer that defines when and how to apply charges. However, instead of listing every group member, the global charge sharing group indicates all accounts in the system or all services of a specific type as members by using a type-only POID.
About Global Charge Sharing Groups

Any existing account or service in your system is automatically considered a part of the group. Likewise, when you create a new account or service in the system, BRM automatically considers the account or service as part of the global charge sharing group.

To create global charge sharing groups, perform the following:

1. Make sure the global charge sharing search is enabled.
   See "Enabling Global Charge Sharing Searches during Discounting".

2. Use Billing Care or a custom client application to create global charge sharing groups. To create global charge sharing groups by using a custom client application, see "Using Third-Party Applications to Manage Global Charge Sharing Groups".

Enabling Global Charge Sharing Searches during Discounting

By default, the global charge sharing search is disabled.

- With this feature enabled, BRM searches for global charge sharing objects during the discounting process.
- With this feature disabled, rating does not search for global charge sharing objects during the discounting process, thereby resulting in better performance.

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

To enable global charge sharing searches during discounting:

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

3. In the XML file, change disabled to enabled:
   
   ```
   <EnableGlobalChargeSharing>enabled</EnableGlobalChargeSharing>
   ```

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

Using Third-Party Applications to Manage Global Charge Sharing Groups

To set up a third-party client application to create, modify, and delete global charge sharing groups, you must customize the application to accept the following

---

**Important:** Charge sharing group members must be either a type-only POID or a list of member POIDs; members cannot include both types.
information and to pass it in the input flist to the Subscription Management FM opcodes:

- Name of the global charge sharing group
- Owner of the group
- Members of the group
- The chargeshare offer associated with the group

To add all accounts in the system as members, you pass a type-only account POID and a NULL service POID in the input flist’s PIN_FLD_MEMBERS array. For example:

```
0 PIN_FLD_MEMBERS ARRAY [0] allocated 2, used 2
1 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account -1 0
1 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.0 /service 0 0
```

To add all services of a specific type in the system as members, you pass a type-only account POID and a type-only service POID in the input flist’s PIN_FLD_MEMBERS array. For example:

```
0 PIN_FLD_MEMBERS ARRAY [0] allocated 2, used 2
1 PIN_FLD_ACCOUNT_OBJ POID [0] 0.0.0.1 /account -1 0
1 PIN_FLD_SERVICE_OBJ POID [0] 0.0.0.1 /service/email -1 0
```

**Note:** In both cases, you pass a type-only account POID in the PIN_FLD_ACCOUNT_OBJ field. The field cannot be NULL.

You pass the information from your client application to the following Subscription Management FM opcodes:

- To create a global charge sharing group, use PCM_OP_SUBSCRIPTION_SHARING_GROUP_CREATE.
- To modify a global charge sharing group, use PCM_OP_SUBSCRIPTION_SHARING_GROUP_MODIFY.
- To delete a global charge sharing group, use PCM_OP_SUBSCRIPTION_SHARING_GROUP_DELETE.

**Note:** For global charge sharing groups, do not call the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP opcode.

For more information about these opcodes, see *BRM Opcode Guide*. 

**About Creating, Modifying, and Deleting Charge and Discount Sharing Groups**

This section covers the following topics:

- About Creating Charge and Discount Sharing Groups
- About Modifying Charge and Discount Sharing Groups
- About Deleting Charge and Discount Sharing Groups
About Creating Charge and Discount Sharing Groups

To create a charge or discount sharing group, you specify the following:

- A group owner account or service
- Member services
- (Discount sharing) The group owner’s discounts to share with the members
- (Charge sharing) The charges, or chargeshare offers, that the group owner assumes for the members

Be aware of the following restrictions:

- Charges or discounts selected for sharing must be valid. A valid charge or discount is one that has an active or inactive status and has not expired.
- The owner of a charge or discount sharing group cannot also belong to a member-owned charge or discount sharing group of the same type.
  
  For example, Anna is the owner of discount sharing group DG1 and wants Sam to be a member. But Sam is the owner of a different discount sharing group, DG2. Anna can have Sam as a member of DG1 only if she is not a member of DG2.
- The name of a charge or discount sharing group must be unique and cannot be used for another sharing group owned by the group owner.
- When adding a service as a member, you can add the service as either a service type or a specific service instance.
  
  If you add a service type as a member, 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.
- All members must have the same primary currency as the owner.

Note: This restriction does not apply to global charge sharing groups.

After you create a charge or discount sharing group, members can join the group. When they join the group, an ordered balance group is created. The ordered balance group determines which sharing groups the member can use and the order in which the groups are used. See “How Charges and Discounts Are Applied”.

You create charge and discount sharing groups in Billing Care.

To implement charge and discount sharing through a custom client application, use the PCM_OP_SUBSCRIPTION_SHARING_GROUP_CREATE opcode. Use the PCM_OP_SUBSCRIPTION_POL_GET_SPONSORS opcode to get a list of chargeshare offers. See BRM Opcode Guide.

When creating a discount sharing group, the moment when sharing starts is determined by a pin.conf entry. See "Configuring the Start and End Times for Discount Sharing”.

About Modifying Charge and Discount Sharing Groups

You can modify a charge or discount sharing group in the following ways:

- Delete members from the group.
When a member is deleted from the group, the member can no longer use discounts from the shared balances or have its charges assumed by the owner. Deleting a member removes the member from the group. It also removes the group from the member’s ordered balance group list. See "How Charges and Discounts Are Applied".

- **Delete chargeshare offers or shared discount offers.**

  When a chargeshare offer is deleted, the owning balance group no longer receives balance impacts for member-generated events that are part of that chargeshare offer. When a shared discount offer is deleted, members can no longer apply that discount, and any owner balances for that discount cannot be used by the members.

  Deleting a chargeshare offer or a shared discount offer removes it from the group. It also removes the group from the members' ordered balance group lists.

- **Add members to the group.**

  When a member is added to the group, that member can begin benefiting from sharing by joining the group. Adding new members updates the group’s members list.

  After you add members to the group, they can join the group. When they join the group, an ordered balance group is created and sharing for that member can begin. See "How Charges and Discounts Are Applied".

- **Add shared charges or discounts.**

  When charges are added to a charge sharing group, the owner account begins to receive the balance impact of those charges from the member services. When discounts are added to a discount sharing group, members can begin to use the discounts for the events generated by their accounts.

  Adding charges or discounts updates the group’s sponsors or discounts list. The charges must be defined as chargeshare offers in the database, and the discounts must be owned by the group owner. The charges and discounts must have an active or inactive status and valid dates.

- **Change the owner of the group.**

  When the owner of a discount sharing group changes, members use discounts from the new owner’s shared balances rather than from the old owner’s balances. When the owner of a charge sharing group changes, the new owner assumes charges generated by the members, and the old owner stops assuming the charges.

  Changing the owner assigns a new group owner, deletes the sponsors or discounts list belonging to the previous owner, and creates a new sponsors or discounts list containing the charges or discounts shared by the new owner.

You modify charge and discount sharing groups in Billing Care.

If you are implementing discount sharing through a third-party client application, you customize the application to invoke BRM opcodes. See BRM Opcode Guide and the following sections:

- About Deleting Charge and Discount Sharing Groups
- About Modifying Charge and Discount Sharing Groups

When members or discounts are added to or deleted from a discount sharing group, the moment when sharing starts and ends is determined by a pin.conf entry. See "Configuring the Start and End Times for Discount Sharing".
About Deleting Charge and Discount Sharing Groups

A charge or discount sharing group is deleted when the group’s owner account is closed or when the sharing group is deleted by a CSR. When a discount sharing group is deleted, members can no longer use the discounts that were shared by the owner. When a charge sharing group is deleted, the owner no longer assumes members’ charges.

When a charge or discount sharing group is deleted, BRM also deletes the sharing group from each member’s ordered balance group, effectively ending the membership. For more information about ordered balance groups, see “How Charges and Discounts Are Applied”.

You delete charge and discount sharing groups in Billing Care.

If you are implementing charge or discount sharing through a third-party client application, use the PCM_OP_SUBSCRIPTION_SHARING_GROUP_DELETE opcode. See BRM Opcode Guide.

When a discount sharing group is deleted, the moment when sharing ends is determined by a pin.conf entry. See ’Configuring the Start and End Times for Discount Sharing’.

How Charges and Discounts Are Applied

An account can purchase discounts, share discounts, and have charges that are sponsored. For events generated by a member service, the shared discounts are used before the sponsored charges are applied to the owning balance group. That way, sponsored charges are applied after usage charges have been discounted. To control the sequence in which shared charges and discounts are applied for member-generated events, BRM maintains an ordered balance group for each member.

About Ordered Balance Groups

Each member of a charge or discount sharing group has an ordered balance group (/ordered_balgrp object). Ordered balance groups are created when members first join sharing groups and are updated each time members join new sharing groups or leave sharing groups.

An ordered balance group contains links to the groups that the member has joined, listed by rank. The ordered list determines the sequence in which the group’s balances are impacted by the member-generated events.

**Note:** Ordered balance groups do not include any global charge sharing groups to which the member belongs. For more information, see "About Global Charge Sharing Groups".

The following example, as illustrated in Figure 10–7, shows how BRM applies charges and discounts for a member’s ordered balance group that references discount sharing group X1 and charge sharing group X2.
How Charges and Discounts Are Applied

Creating and Managing Charge and Discount Sharing Groups

Figure 10–7  BRM Application of Discounts and Charges to Ordered Balance Group

BRM applies discounts and charges

```
/ordered_balg
/group/sharing/discounts X1: 1
/group/sharing/discounts -1 (member's own): 2
/group/sharing/charges X2: 3
```

Charges remaining to member

Note: In this example, the entry indicated by -1 references discounts owned by the member.

As shown in Figure 10–7, when the ordered balance group contains both charge and discount sharing groups, the discount sharing group’s balances are used first. This enables discounts to be applied before usage charges are determined for charge sharing. If multiple charge or discount sharing groups are configured, they are applied in the order of their rank.

In addition, BRM always references any charges owned by the member after referencing the ordered balance group. That way, members receive charge sharing benefits before having to use their own charges.

The following example, illustrated in Figure 10–8, shows how a member’s ordered balance group is used to impact the balances of the sharing groups. In this example, Account A is the owner of discount sharing group X1. Account B is the owner of charge sharing group X2. Service A is a member of both X1 and X2. Account A shares a discount that grants 20 minutes of telephone calls monthly. Account B sponsors 50% of usage charges. Service A has two discounts, one for 30 minutes monthly and another that discounts 10% of the charges.
How Charges and Discounts Are Applied

Figure 10–8  Ordered Balance Group Impact on Charge and Discount Sharing Groups

Here, BRM applies shared charges and discounts for events generated by Service A as follows:

1. Service A generates an event for 100 minutes of telephone calls at $0.10 per minute.
   Before any discounts or sponsored charges are applied, Service A owes $10.00.

2. Account A’s discounts are consumed first.
   After consuming 20 minutes, Service A owes $8.00 for 80 minutes of telephone calls at $0.10 per minute.

3. Service A’s discounts are consumed next.
   After consuming 30 minutes, Service A owes $5.00 for 50 minutes of calls at $0.10 per minute. After the 10% usage discount, Service A owes $4.50 for 50 minutes of calls.

4. Account B’s charges are consumed last.
   After a 50% of usage sponsored by Account B, Account B owes $2.25 and Service A owes $2.25.
How Discounts Are Applied When a Member Belongs to the Group Owner Account

When a member of a discount sharing group belongs to the discount sharing group owner account, the discount at the account level is applied twice for the member.

For example, Account A is the owner of discount sharing group X. Account A owns Service A and Service A is also a member of the discount sharing group X. Account A shares a discount that grants 20 minutes. When Service A generates an event for 100 minutes, the event is discounted as follows:

- Service A is granted 20 minutes as a member of the discount sharing group X.
- Because Service A also belongs to the account and any discount purchased at the account level applies to its services, the account discount is applied and Service A is granted an additional 20 minutes.

Creating, Deleting, and Modifying Ordered Balance Groups

Ordered balance groups are created for a member service that participates in charge or discount sharing. A member account can have multiple ordered balance groups depending on how many services are included in sharing groups. For example, if a member account participates in sharing for its IP, email, and GSM services, it has ordered balance groups for each of these services.

Creating Ordered Balance Groups

To create an ordered balance group, you specify the charge and discount sharing groups that provide shared balances for the service or account.

You create ordered balance groups when a member first joins charge or discount sharing groups for a service.

To implement charge or discount sharing through a third-party client application, use the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP opcode. See BRM Opcode Guide.

Deleting Ordered Balance Groups

To delete an ordered balance group, you specify the ordered balance group that you want to remove from the database.

To delete an ordered balance group when implementing charge or discount sharing through a third-party client application, use the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP opcode. See BRM Opcode Guide.

Modifying Ordered Balance Groups

Modifying an ordered balance group consists of adding or deleting charge or discount sharing groups:

- **Adding a sharing group**: A link to the charge or discount sharing group is added to the ordered balance group list.

- **Deleting a sharing group**: The link to the charge or discount sharing group is removed from the ordered balance group list.

**Important**: If you add or delete charge or discount sharing groups, you might need to rearrange the ordered balance group list. See “Modifying the Order in Which Charge and Discount Sharing Groups Are Used”.
If you are implementing charge or discount sharing through a third-party client application, use the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP opcode. See BRM Opcode Guide.

**Modifying the Order in Which Charge and Discount Sharing Groups Are Used**
The sequence of the ordered balance group list determines which shared balances are used first. You can change the order of this list to reprioritize sharing for a member.

You modify the order of the ordered balance group list when you join charge and discount sharing groups or when a member decides to use one charge or discount sharing group before another.

If you are implementing charge or discount sharing through a third-party client application, use the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP opcode. See BRM Opcode Guide.

**Creating or Modifying Multiple Ordered Balance Groups Simultaneously**
To create or modify multiple ordered balance groups simultaneously, you specify the charge and discount sharing group that provides the shared balance group. You also specify a list of member services to share this balance group.

If you are implementing charge or discount sharing through a third-party client application, use the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP_BULK_MODIFY opcode. See BRM Opcode Guide.
About Transferring Rollover Balances

This document describes rollover transfers and explains how to configure Oracle Communications Billing and Revenue Management (BRM) for rollover transfers.

Topics in this document:

■ About Allowing Customers to Transfer Rolled-Over Balances
■ About the Rollover-Transfer Profile
■ About Rerating the Receiver’s Account due to Delayed Billing
■ Configuring BRM to Use Rollover Transfers

See also:

■ Managing Balances
■ List of Accounts Receivable Features

About Allowing Customers to Transfer Rolled-Over Balances

Balances rolled over from one billing cycle to another are usually rolled over within the same account. By using the rollover-transfer feature, you can transfer noncurrency rollover balances from one account or service instance to other accounts or service instances.

BRM can perform rollover transfers in the following ways:

■ From one account to other accounts.
■ From a service instance in one account to service instances in other accounts.
■ From one account to service instances in other accounts.
■ From a service instance in one account to other accounts.
■ From a service instance in one account to other service instances in the same account.
■ Between services, even when the service types differ.

The sender’s service type does not need to match the receiver’s service type as long as both services own the same type of balance, such as dollars or minutes.

BRM cannot transfer rolled over balances in the following circumstances:

■ A member service shares a balance group with its subscription service.
■ The buckets were created as a result of a previous rollover transfer. Previously rolled over balances cannot be rolled over again.
If you offer subscription services, you can transfer rolled-over balances from a subscription service to a member service only if they belong to different balance groups. A member service that shares the same balance group as its subscription service can receive a rollover transfer but cannot send a rollover transfer. For more information about subscription services, see "About Subscription Services" in BRM Managing Customers.

About the Rollover-Transfer Profile

You specify how to transfer rollovers by using a rollover-transfer profile. The rollover-transfer profile specifies the following:

- The account or service that owns the balance grant (the sender)
- The list of balances to roll over
- The accounts or services into which the rollover is transferred (the receiver)
- (Optional) The validity periods in which the rollovers can transfer

You create, modify, and delete rollover-transfer profiles in Customer Center.

When you create a rollover-transfer profile, BRM, by default, verifies that the profile does not have overlapping validity periods or receivers for a given service. You can add or modify the verification checks by customizing the PCM_OP_CUST_POL_VALIDATE_PROFILE policy opcode.

You can select any noncurrency balance to configure a rollover transfer unless there is a rollover rule defined in a charge offer and the sender has not purchased that charge offer. The sender must own the charge offer in which the rollover rule is defined for the rollover transfer to occur.

About Rerating the Receiver’s Account due to Delayed Billing

When delayed billing is configured, balances rolled over to the receiver during the initial billing process can be accessed by delayed events of the sender.

After a rollover transfer, if the sender account is rerated and the rollover buckets are adjusted, the receiver’s rollover amount is also affected. When a bill is finalized for the sender, BRM generates a rollover correction event to adjust the rollover of the sender if any rolled-over balances were consumed by delayed events. BRM adjusts the rollover transfer amount of the receiver accordingly.

See the discussion of rerating in ECE Implementation Guide for more information.

---

**Important:** You must manually rerate the receiver’s account if events rated before the rollover correction were consumed from the rollover-transfer amount. BRM does not automatically create rerate jobs for the receiver.

---

**Note:** The receiver must be rerated if there is rollover correction for the sender, even when the sender is not rerated.

---

Configuring BRM to Use Rollover Transfers

Before you can use rollover transfers, you must perform the following tasks:
1. Enable rollover transfers in the `/config/business_params` object.

2. Set up event notification for rollover transfers.

To enable rollover transfers, run the `pin_bus_params` utility to change the `RolloverTransfer` business parameter. For information about this utility, see *BRM Developer’s Guide*.

1. Go to `BRM_home/sys/data/config`.

2. Create an XML file from the `/config/business_params` object:

   ```
   pin_bus_params -r -c "Subscription" bus_params_subscription.xml
   ```

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

   ```
   <RolloverTransfer>enabled</RolloverTransfer>
   ```

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*.

To configure event notification for rollover transfers, do the following:

1. Depending on which BRM features you use, your system may contain one or more configuration files for event notification.

   If your system contains more than one of these files, you must merge their contents into a single file.

   All of the event notification files available in your system are in the `BRM_home/sys/data/config` directory.

2. Add the following entries to the event notification file:

   ```
   # Event notification for rollover transfers
   9069 0 /event/billing/cycle/rollover/monthly
   9069 0 /event/billing/cycle/rollover_correction
   ```

   This configures the `/event/billing/cycle/rollover/monthly` and `/event/billing/cycle/rollover_correction` events to call the `PCM_OP_SUBSCRIPTION_TRANSFER_ROLLOVER` opcode (opcode ID number 9069) to determine if a rollover transfer can occur.

   See *BRM Opcode Guide*.

3. (Optional) If necessary, add, modify, or delete entries in your event notification file.

4. (Optional) If necessary, create custom code for event notification to trigger.

5. Load your event notification file into the BRM database’s `/config/notify` object by running the `load_pin_notify` utility (see *BRM Managing Customers*).

For more information, see "Using Event Notification" in *BRM Developer’s Guide*. 
Managing Balances

This document describes how Oracle Communications Billing and Revenue Management (BRM) stores, displays, and manipulates balance data.

Topics in this document:
- Transferring Balance Amounts
- Moving a Balance Group from One Bill Unit to Another
- Backdating A/R Actions
- Synchronizing BRM and ECE Balance Group Transfer Data

See also:
- About Transferring Rollover Balances
- List of Accounts Receivable Features
- BRM Concepts

Transferring Balance Amounts

Transfers are made to correct payments that were allocated incorrectly, to allocate a debit adjustment for any account, and to allocate an account adjustment for an account that pays by credit card. In Customer Center, a CSR can transfer amounts between bill items. Transfers can also be made from payment items and account adjustment items to bill items.

A CSR can make a credit or debit transfer of a currency balance.

To customize how to transfer balance amounts between items, use PCM_OP_BILL_TRANSFER_BALANCE. To change how BRM validates amounts being transferred, customize CM_OP_BILL_POL_VALID_TRANSFER. See BRM Opcode Guide.

You can transfer an amount from a bill that has a negative balance to one or more bills that have a positive balance. To do so, create a custom application that calls PCM_OP_AR_BILL_CREDIT_TRANSFER. See BRM Opcode Guide.

Moving a Balance Group from One Bill Unit to Another

To move a balance group from one bill unit to another in the same account, use Billing Care.

Moving a balance group to a different bill unit means that any new charges for the services in the balance group are applied to the new bill unit.
For example, an account has two bill units. One bill unit tracks charges for services that are invoiced. The other tracks charges for services paid by credit card. The customer decides to pay for all services by credit card. The balance group for invoiced services is moved to the bill unit with the credit card payment method. Any new charges for these services are applied to the new bill unit and are charged to the credit card. Existing charges for these services that occurred before the balance group was moved are associated with the old bill unit and are invoiced.

You cannot move a balance group to another bill unit if the balance group’s bill unit has unallocated payments or adjustments, open refunds, or unresolved disputes. All disputes must be settled, refunds paid, and payments and adjustments allocated before the balance group can be moved.

A bill unit must have at least one balance group. When a bill unit has only one balance group and that balance group is moved to another bill unit, BRM automatically creates a new balance group not associated with any service for the bill unit from which the balance group was moved.

Important:

■ You cannot move account default balance groups to a different bill unit.

■ You should never move a balance group to a bill unit in a different account. To have a different account be responsible for the charges in a balance group, you must create a bill unit hierarchy that includes the appropriate bill units from both accounts.

Note: To move a balance group from one bill unit to another in the same account, create a custom application that calls PCM_OP_CUST_MODIFY_CUSTOMER.

Backdating A/R Actions

You can backdate A/R actions so that, for accounting purposes, the action is considered to have occurred at an earlier point in time and the revenue for that time is reported accurately. The A/R actions you can backdate include adjustments, disputes, settlements, externally initiated payments, payment reversals, and write-offs.

Note: BRM does not support future dating of A/R actions.

To backdate an A/R action, the CSR sets the transaction date (the date that the action is to take effect) to a date before the current date. When backdating, the CSR must select a date after the following:

■ The last posted G/L transaction report. See "Posting G/L Reports" in BRM Collecting General Ledger Data.

■ The account creation date

Whether backdated A/R actions appear on the current bill depends on the accounting method you use:

■ If you use balance forward accounting, the total of A/R actions for all prior open bills appears on the current bill or invoice as part of the previous balance.

■ If you use open item accounting, A/R actions do not appear on the current bill or invoice.
If customers request an invoice that includes information on specific A/R actions in a prior billing period, you can create the invoice by running PCM_OP_INV_MAKE_INVOICE.

**Synchronizing BRM and ECE Balance Group Transfer Data**

You must configure BRM to synchronize the following database updates with the ECE cache:

- Service transfers to a different balance group
- Balance group transfers to a different bill unit

To configure BRM to send updated balance group transfer information to ECE, use the account synchronization feature to publish the `ServiceBalanceGroupTransfer` business event to the Oracle DM database queue. For installation and configuration instructions, see "Installing and Configuring Account Synchronization" in *BRM Installation Guide*.

When configuring the Oracle DM for account synchronization:

- Make sure the `ServiceBalanceGroupTransfer` business event is listed in your payload configuration file under the `<PublisherDefs>` section. This business event appears in the default Oracle DM payload configuration file (`BRM_home/sys/eai_js/payloadconfig_ifw_sync.xml`).

- Make sure the event notification list maps the `/event/notification/service_balgrp_transfer/data` notification event to opcode number 3626. This mapping appears in the default account synchronization event notification file (`BRM_home/sys/data/config/pin_notify_ifw_sync`).

- Add the `ServiceBalanceGroupTransfer` business event to your `BRM_home/sys//dm_oracle/ifw_sync_queuenames` file if the file maps specific business events to queues.
This chapter describes Oracle Communications Billing and Revenue Management (BRM) Balance Monitoring Manager.

About Balance Monitoring

You use balance monitoring to enable your customers to monitor the balance of a group of accounts in near real time. Balance monitoring collects the balance impacts for a specified group of accounts and services, and notifies customers when their balance is too high. Customers and CSRs can also access the group’s total balance at any time by using a custom client interface.

For example, a family with multiple wireless telephony accounts can create a balance monitor that tracks the entire family’s balance and that alerts them when the balance total exceeds $100.

Balance monitoring provides advantages to:

- **Service providers** because it reduces the risk of debt exposure from nonpaying customers. Customers who track their balances are less likely to accrue excessive charges that they cannot pay.
- **Customers** because they can monitor spending habits and are alerted when charges exceed their specified values. They can then reduce usage or inform members of the group that are generating excessive charges.

To create a balance monitor, customers define:

- All account and service balances they want to include in their balance monitor. This group of accounts and services forms a *monitor group*. See "About Monitor Groups".
- When to be alerted that the balance total is too high or approaching the maximum. See "Alerting Customers When Monitored Balances Cross Limits or Thresholds".

About Monitor Groups

Customers specify which account or service balances they want to track by creating a monitor group. Each monitor group includes the following:

- **The owner**. See "About Monitor Group Owners".
- **All members**. See "About Monitor Group Members".
- **The monitor group type**. See "About Monitor Group Types".
About Monitor Groups

Customers can create monitor groups manually or automatically:

- With manual creation, customers specify each member to include in the monitor group.
- With automated creation, customers only specify an owner and monitor type, and BRM automatically finds and adds members to the monitor group.

See "About Creating and Maintaining Balance Monitors".

About Monitor Group Owners

Monitor group owners have permission to view group balances, add or remove group members, and add or change threshold and credit limit values.

Monitor group owners can be accounts or services and can be anywhere in a hierarchy chain. For example, an owner could be a parent account in an account hierarchy or a paying or nonpaying child account.

About Monitor Group Members

Any of the following can be members of a monitor group:

- **Accounts.** Members of a monitor group can include parent accounts, paying child accounts, and nonpaying child accounts. BRM monitors the balances for all services under a member account. For example, if a member account contains a GSM service and a GPRS service, BRM adds the current balance for both services to the group balance.

- **Services.** Members of a monitor group can include services, such as the GSM service, for a group of accounts. When the member is a service, BRM monitors charges for that service only. For example, a company that pays all GSM services for its employees could create a monitor group that includes each employee’s GSM service but excludes any other services owned by the employee accounts.

Accounts and services can belong to one or more monitor groups. BRM tracks the monitor groups to which an account or service belongs and then accesses this information during the rating process to determine where to apply balance impacts.

About Monitor Group Types

Monitor group types specify the type of members a monitor group contains. For example, group membership can be restricted to nonpaying child accounts and their services, or it can include both paying and nonpaying child accounts and their services.

Table 13–1 shows the monitor group types supported by BRM:

<table>
<thead>
<tr>
<th>Monitor Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hierarchy</td>
<td>Includes paying and nonpaying child accounts and their services.</td>
</tr>
<tr>
<td>Paying Responsibility</td>
<td>Includes nonpaying child accounts and their services.</td>
</tr>
<tr>
<td>Service Level</td>
<td>Includes a member service in a subscription group.</td>
</tr>
</tbody>
</table>

When a customer attempts to add a member to a monitor group, BRM validates the member against rules you specify for each monitor group type. The default implementation contains no validation rules for these monitor group types, but you
can create custom group types or create your own rules by customizing the PCM_OP_SUBSCRIPTION_POL_PREP_MEMBERS policy opcode. For information, see "Validating the Members of a Balance Monitor Group".

### About the Balances of a Monitor Group

The balance for a monitor group is a running total that includes all balance impacts generated by its members. BRM updates the balance when a group member generates a usage event or when you create or modify a monitor group.

BRM stores the total balance for a particular monitor group in /balance_group/monitor objects in the database. For information, see "Creating /balance_group/monitor Objects".

#### Balance Impacts Included in a Monitored Balance

Monitored balance totals include the following:

- **All standard Accounts Receivable (A/R) impacts**, such as these:
  - Discounts
  - Account-level adjustments and disputes
  - Payments
  - Refunds
  - Write-offs

  For more information, see "Managing Balances".

- **Taxes.** Monitored balance totals include taxes that are applied during the real-time rating process only. Any taxation that is deferred to the billing process is not included in the balance total.

- **Sponsored or charge sharing charges.** Monitored balance totals include charges that apply to the sponsor’s account only.

  See "About Monitoring Charge Sharing and Sponsored Account Balances".

### About Monitoring Charge Sharing and Sponsored Account Balances

When an account belongs to a charge sharing or sponsor group, its balances may be paid by more than one account. For example, an employee may pay his own wireless usage fees but have his monthly subscription fees paid by his company. When this occurs, each balance monitor tracks only the charges for which it is responsible. In this example, the company’s balance monitor tracks the subscription fee, and the employee’s balance monitor tracks all usage charges.

### When Balance Impacts Are Added to a Monitored Balance

BRM updates the balance of a monitor group at the following times:

- **When a balance monitor is created.** When you first create a balance monitor and run the `pin_monitor_balance` utility, BRM automatically calculates the group balance. BRM retrieves the current balance of each member and adds it to the group balance.

- **When members are added to or removed from a monitor group.** When a member is added to a monitor group, BRM automatically adds the member’s current balance to the group balance when `pin_monitor_balance` is run. Likewise, when a
Alerting Customers When Monitored Balances Cross Limits or Thresholds

When a member is removed from a monitor group and `pin_monitor_balance` is run, BRM automatically decreases the group balance.

- **As a result cycle-fee rating.** When rating events, BRM calculates any monitor balance impacts immediately. However, the impacts are not added to the group balance until you run the balance monitor utility. Therefore, there is a small lag between the time an event occurs and the time it impacts the group balance.

### Alerting Customers When Monitored Balances Cross Limits or Thresholds

To be alerted when the balance of their monitor group is approaching predefined values, customers must set up thresholds and credit limits. BRM then tracks the group balances and generates notification events when the balances cross above or below the threshold value or credit limit.

BRM checks whether a group balance has crossed a threshold or a credit limit whenever the following occur:

- The balance of a monitor group is updated
- Monitor group owners add or change limit and threshold values

### About Setting Limits and Thresholds

To be alerted when a monitored balance reaches a certain value, customers must define the following for each balance monitor:

- **Credit floor.** The credit floor specifies the starting point for a threshold value. The default credit floor is NULL. You must set the credit floor to a non-NULL value.

- **Credit limit.** The credit limit specifies the ending point for a threshold value. BRM uses the credit limit to calculate the threshold value, which is a percentage of the credit limit. BRM generates a notification event when the group balance crosses above or below the credit limit.

---

**Note:** Balance monitor credit limits are separate from account credit limits. Account credit limits specify the maximum balance for an individual account and affect the way BRM handles authorizations and rating. Monitor credit limits are used only to calculate threshold values and do not affect authorization and rating.

---

- **Thresholds.** Thresholds specify the balance totals that trigger an alert to CSRs and monitor group owners. Owners specify thresholds in the following ways:

  - As a percentage of the credit limit in 5% increments (75%, 80%, 85%, and so on). To be notified when the credit limit is reached, select the 100% threshold setting.

  - As a fixed value, such as $90 or 50 minutes.

**Important:** You must set a value for each of these parameters. BRM cannot send notification events when any of the following are true: the credit floor is NULL, the credit limit is infinite or undefined, or the threshold is undefined.
For example, a customer might set a monitor group’s credit floor to $0, credit limit to $100, and thresholds to 70%, 90%, and 100%. This causes BRM to generate notification events whenever the monitored balance crosses above or below $70, $90, and $100.

You define a balance monitor’s credit floor, credit limit, and threshold values when you create the balance monitor in a custom client application. BRM then stores the values in a /config/credit_profile object in the BRM database.

About Using Event Notification to Alert Customers

BRM uses event notification to alert monitor owners when balances cross a threshold or credit limit. When a monitored balance crosses a threshold or limit, the following occurs:

1. BRM generates one of these notification events:
   - /event/notification/threshold — Indicates that the balance crossed above a threshold value or credit limit.
   - /event/notification/threshold_below — Indicates that the balance crossed below a threshold value or credit limit.

   For information about these events, see "Event Notification Definitions" in BRM Developer’s Reference.

2. BRM calls the opcode associated with the event in your system’s event notification list. See "About the Event Notification List" in BRM Developer’s Guide.

3. The opcode processes the event data and passes information to your custom client application.

To enable this, you must configure the event notification feature, the opcode (if it is a policy opcode), and your custom client application to process these events and alert balance monitor owners. For more information, see "Configuring Event Notification for Balance Monitoring".

About Notification Events for Balance Monitoring

BRM generates notification events whenever a monitored balance crosses over or under a credit limit or threshold. This enables you to make BRM perform different actions depending on the event type. For example, you can customize BRM to send an email to both you and the group owner when the group balance crosses above a threshold, but to send an email only to the group owner when the balance falls below a threshold.

About the Number of Notification Events

BRM generates one notification event for each monitored balance that crosses a threshold or credit limit. When one event causes a balance to cross multiple thresholds, BRM generates only one event that lists all settings that were breached.

For example, member A belongs to balance monitors 1 and 2. When member A generates a usage event that crosses the 50% and 55% thresholds for monitor 1, and the 40% threshold for monitor 2, BRM generates only two events:

- The first notification event specifies that the balance of monitor 1 surpassed both the 50% and 55% thresholds.
- The second notification event specifies that the balance of monitor 2 surpassed the 40% threshold.
Information Included in Balance Monitor Notification Events

Each balance monitor notification event includes the following information:

■ Target balance monitor.
■ Resource ID.
■ Alert type: Credit limit or threshold breach.
■ Reason for the breach:
  – **Upward breach.** The monitored balance crossed above a threshold value.
  – **Downward breach.** The monitored balance crossed below a threshold value, for example, when the customer makes a payment.
  – **Upward reset.** The owner of the balance monitor increased a threshold value, causing the balance to cross below a threshold. For example, this occurs when the current balance is $50 and you change the value of threshold 1 from $40 to $60. In this case, the $50 balance crosses below the new setting for threshold 1.
  – **Downward reset.** The owner of the balance monitor decreased a threshold value, causing the balance to pass above a threshold. For example, this occurs when the current balance is $40 and you change the value of threshold 1 from $50 to $25. In this case, the $40 balance passes above the new setting for threshold 1.
■ Monitor type: Hierarchy credit exposure, paying responsibility credit exposure, or service level.
■ Source of the breach.

---

**Note:** BRM does not provide a source ID when the alert is due to a threshold reset or group member addition.

■ List of thresholds or credit limits that were breached. For example, 50% and 70%. For credit limit breaches, this field is set to 100%.

For more information about these events, see "Updating Monitor Balances and Sending Credit Limit/Threshold Breach Notifications".

Providing Real-Time Access to the Balances of Monitor Groups

Balance monitor owners can access the balance for their monitor groups in real-time by using a Web interface, such as a Web-enabled phone or computer. When monitor owners log in, BRM collects their account information and optionally a date range and then retrieves the following for each balance monitor:

■ Beginning balance of the monitor group on the start date
■ Ending balance of the monitor group on the end date

To provide your customers and CSRs with access to real-time monitor balances, you must customize your client interface to use the BRM API. For information, see "Displaying Balance Monitor Information in Client Applications".

About Creating and Maintaining Balance Monitors

You create and modify balance monitors by using a third-party customer relationship management (CRM) application. To implement balance monitoring in your CRM
application, you must configure it to accept the following information for each balance monitor and pass it to the BRM API:

- Balance monitor name
- Balance monitor owner
- List of group members
- The monitor group type: hierarchy, paying responsibility, or service level
- Credit limit (optional)
- Credit floor (optional)
- Threshold settings (optional)

**Balance Monitor Creation Process Overview**

The process that BRM uses to create or modify a balance monitor depends on whether you implement balance monitoring in your system with or without Automated Monitor Setup (AMS).

- **With AMS**, you can automate many of the balance monitor processes. AMS automatically adds and removes members from the balance monitor for you.

- **Without AMS**, you add and remove members manually. You must use this method to create balance monitors for custom monitor types.

BRM creates balance monitors as follows:

1. Takes as input all information about the balance monitor, such as the monitor owner and threshold settings.
2. Creates a bucket to store the balance of the monitor group.
3. Adds members to the balance monitor. BRM determines whether the following is true:
   - AMS is enabled.
   - The monitor type is hierarchy, paying responsibility, or service level.

   **If both are true**, BRM uses AMS to add members. BRM uses the monitor owner and monitor type to find all appropriate child accounts and services, and adds them to the monitor. See "About Using AMS to Manage Balance Monitors Automatically".

   **If either one is false**, BRM adds members without AMS. BRM takes as input the list of members and validates them against custom criteria. All members that pass validation are added to the balance monitor. See "Managing Balance Monitors without AMS".

4. Retrieves the current balance of all members and adds it to the group balance.

**About Using AMS to Manage Balance Monitors Automatically**

You configure BRM to automatically add and remove members from balance monitors, based on the monitor group type, by using Automated Monitor Setup (AMS). AMS is a group of opcodes that automate many of the balance monitor processes, making balance monitoring easier to set up and use.

AMS creates and updates balance monitors by following the organizational hierarchy of accounts and services. This synchronizes the membership between a hierarchy and a monitor group.
About Creating and Maintaining Balance Monitors

**Important:** AMS works only for the three default monitor group types: hierarchy, paying responsibility, and service level. For custom group types, you must create balance monitors without AMS. See "Managing Balance Monitors without AMS".

AMS automatically adds and removes members from balance monitors when the following occur:

- **A balance monitor is created.** AMS takes as input the parent (owner) account or service and the monitor type, and then searches through the parent’s lineage to find accounts and services that match the criteria for the monitor type. All subordinate accounts and services that meet the criteria are added as members.

  For example, when users create a paying responsibility-type monitor, AMS adds to the monitor the parent account and its services and all subordinate nonpaying child accounts and their services.

- **An account hierarchy or subscription group is changed.** AMS automatically adds or removes members from any associated balance monitors, based on the monitor type. For example, when you add a nonpaying child account to a hierarchy, AMS automatically:
  - Finds all parent accounts at a higher level in the hierarchy than the specified child account.
  - Finds all hierarchy and paying responsibility balance monitors associated with each parent account.
  - Adds the child account and its services to each balance monitor.

When a balance monitor is created or a hierarchy or subscription group changes, the following occurs:

1. BRM generates an AMS notification event.
2. The BRM event notification feature calls one of the AMS opcodes.
3. The AMS opcodes add or remove members from the appropriate monitor group and update the monitored balance.

**Managing Balance Monitors without AMS**

When creating or updating balance monitors without AMS, users are responsible for adding and removing members. BRM does not verify that balance monitors include all appropriate child accounts and services in a hierarchy or subscription group nor does it check whether a hierarchy or subscription group changes over time.

Users must add or remove members when the following occur:

- **A balance monitor is created.** Users specify the individual accounts and services to add to the balance monitor.

- **An account hierarchy or subscription group is changed.** Users must manually update any balance monitors associated with the hierarchy or subscription group. When an account is added to a hierarchy, users must manually add the account to all balance monitors associated with the group. Likewise, when an account is removed from a hierarchy, users must manually remove the account from any associated monitors.

To manage balance monitors without AMS, see "Using the Balance Monitor API to Create and Maintain Balance Monitors".
Balance Monitoring Process Overview

BRM processes balance monitoring data as follows:

1. BRM rates events and collects any monitored balance impact data.
2. BRM publishes monitored balance impacts to the monitor queue.
   
   
   See "Understanding the Monitor Queue".
3. The pin_monitor_balance utility retrieves data from the monitor queue and calls
   the PCM_OP_BAL_APPLY_MONITOR_IMPACTS opcode to perform the
   following:
   
   ■ Add balance impacts to the appropriate monitored balance in the BRM
     database. See "Using pin_monitor_balance to Update Monitored Balances".
   ■ Determine whether the monitored balance crossed any thresholds and, if it
ev   did, generate a notification event. See "About Using Event Notification to
   Alert Customers".

About Balance Monitoring and Real-Time Rating

Real-time rating processes monitored balances as follows:

1. Rates the event and applies any discounts.
2. Determines whether the event owner belongs to any monitor groups.
3. Determines whether balance monitoring is enabled.
4. Generates monitor balance impacts for each monitor group listed in the /ordered_
balgrp object. The monitor balance impact includes the following data:
   ■ Event owner
   ■ Amount
   ■ Resource ID
   ■ Balance monitor (POID of the /balance_group/monitor object)
5. Publishes the monitor balance impacts to the monitor queue.
   
   See "Understanding the Monitor Queue".

Understanding the Monitor Queue

The monitor queue (/monitor_queue) holds all monitor balance impacts generated by
rating. The pin_monitor_balance utility retrieves this information and updates the
balances of the associated monitor groups.

For performance reasons, the pin_monitor_balance utility does not delete /monitor_
queue objects after they are retrieved from the BRM database. Instead, the utility flags
the /monitor_queue objects as "processed" to prevent the objects from being consumed
again.

BRM tracks the status of /monitor_queue objects by using the PIN_FLD_STATUS field:

■ When the field is set to 0, the object has not been processed by the pin_monitor_
  balance utility.
■ When the field is set to a nonzero value, the object has been processed by the
  utility.
Using pin_monitor_balance to Update Monitored Balances

You use the **pin_monitor_balance** utility to apply monitor balance impacts to a monitored balance. This utility is a multithreaded application (MTA) that retrieves a list of all `/monitor_queue` objects in the BRM database and then spawns child threads to process each object individually.

You can configure the utility to run with one thread or multiple threads:

- One thread enables the utility to process events in the order that they occurred but decreases processing performance.
- Multiple threads increase performance but may cause the utility to process events out of order. The default is 5 threads.

To configure the number of threads, see "Configuring pin_monitor_balance to Process Events in the Order Created".

You can run `pin_monitor_balance` manually or configure a scheduler, such as `cron`, to run it at predefined intervals. For more information, see "Running pin_monitor_balance to Update Monitored Balances".

The **pin_monitor_balance** utility performs the following tasks:

1. Retrieves all unprocessed `/monitor_queue` objects from the BRM database (that is, all objects with the PIN_FLD_STATUS field set to 0).
2. Retrieves the `/event` object associated with the `/monitor_queue` object to obtain the following:
   - Monitor balance impacts
   - Sub-balance impacts
   - Target balance monitors (/balance_group/monitor objects)
   - Event owner
3. Calls the PCM_OP_BAL_APPLY_MONITOR_IMPACTS opcode, which performs the following:
   - Updates the monitored balance (/balance_group/monitor).
   - Determines whether any balance monitor thresholds have been breached by comparing the monitored balance to the threshold and credit limit values stored in the `/config/credit_profile` object. If any thresholds or limits were breached, the opcode generates a notification event. See "About Using Event Notification to Alert Customers".
   - Sets the `/monitor_queue` object’s PIN_FLD_STATUS field to a nonzero value to prevent it from being processed again by the utility.

Configuring Your BRM System for Balance Monitoring

To configure your system for balance monitoring, perform the following tasks:

1. Enable balance monitoring in BRM.
   
   See "Enabling Balance Monitoring in BRM".
2. Enable AMS in BRM.
   
   See "Enabling AMS in BRM".
3. Configure BRM to perform appropriate follow-up operations when limits and thresholds are crossed.
See "Configuring Event Notification for Balance Monitoring".

4. Configure pin_rerate to wait a specified amount of time before starting the rerating process.

See "Specifying a Wait Time before Rerating Events".

5. (Optional) Configure pin_monitor_balance to process monitor balance impacts in the order that they were created.

See "Configuring pin_monitor_balance to Process Events in the Order Created".

Enabling Balance Monitoring in BRM

By default, balance monitoring is disabled in BRM. You can enable this feature by modifying a field in the multi-bal instance of the /config/business_params object.

When balance monitoring is enabled, BRM monitors currency resources.

You modify the /config/business_params object by using the pin_bus_params utility (see "pin_bus_params" in BRM Developer’s Guide).

To enable balance monitoring:

1. Use the following command to create an editable XML file from the multi-bal instance of the /config/business_params object:

   pin_bus_params -r BusParamsMultiBal bus_params_multi_bal.xml

   This command creates the XML file named bus_params_multi_bal.xml.out in your working directory. If you do not want this file in your working directory, specify the path as part of the file name.

2. Search the XML file for following line:

   <BalanceMonitoring>disabled</BalanceMonitoring>

3. Change disabled to enabled.

   Caution: BRM uses the XML in this file to overwrite the existing multi-bal instance of the /config/business_params object. If you delete or modify any other parameters in the file, these changes affect the associated aspects of the BRM balance monitoring configuration.

4. Save the file and change the file name from bus_params_multi_bal.xml.out to bus_params_multi_bal.xml.

5. Use the following command to load this change into the /config/business_params object:

   pin_bus_params bus_params_multi_bal.xml

   You should execute this command from the BRM_Homesys/data/config directory, which includes support files used by the utility. To execute it from a different directory, see "pin_bus_params" in BRM Developer’s Guide.

6. Read the object with the testnap utility or the Object Browser to verify that all fields are correct.

   For general instructions on using testnap, see "Using testnap" in BRM Developer’s Guide. For information on how to use Object Browser, see "Reading Objects by Using Object Browser" in BRM Developer’s Guide.
7. Stop and restart the Connection Manager (CM).
   For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

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

### Enabling AMS in BRM

By default, AMS is disabled in BRM. You can enable this feature to automate many of the balance monitoring processes by modifying a field in the subscription instance of the `/config/business_params` object.

You modify the `/config/business_params` object by using the `pin_bus_params` utility (see "pin_bus_params" in *BRM Developer’s Guide*).

To enable AMS:

1. Use the following command to create an editable XML file from the subscription instance of the `/config/business_params` object:
   
   ```bash
   pin_bus_params -r BusParamsSubscription bus_params_subscription.xml
   ```
   
   This command creates the XML file named `bus_params_subscription.xml.out` in your working directory. If you do not want this file in your working directory, specify the path as part of the file name.

2. Search the XML file for following line:
   
   ```xml
   <AutomatedMonitorSetup>disabled</AutomatedMonitorSetup>
   ```

3. Change `disabled` to `enabled`.

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

4. Save the file and change the file name from `bus_params_subscription.xml.out` to `bus_params_subscription.xml`.

5. Use the following command to load this change into the `/config/business_params` object:
   
   ```bash
   pin_bus_params bus_params_subscription.xml
   ```
   
   You should execute this command from the `BRM_Home/sys/data/config` directory, which includes support files used by the utility. To execute it from a different directory, see "pin_bus_params" in *BRM Developer’s Guide*.

6. Read the object with the `testnap` utility or the Object Browser to verify that all fields are correct.

   For general instructions on using `testnap`, see "Using testnap" in *BRM Developer’s Guide*. For information on how to use Object Browser, see "Reading Objects by Using Object Browser" in *BRM Developer’s Guide*.

7. Stop and restart the Connection Manager (CM).
Configuring Your BRM System for Balance Monitoring

For more information, see "Starting and Stopping the BRM System" in BRM System Administrator’s Guide.

8. (Multischema systems only) Run the pin_multidb script with the -R CONFIG parameter.

For more information, see "pin_multidb" in BRM System Administrator’s Guide.

Configuring Event Notification for Balance Monitoring

When the balance of a monitor group crosses a threshold or a credit limit, BRM generates the following notification events. You must configure the BRM event notification feature to call opcodes that perform the appropriate follow-up operations when these events are generated.

- /event/notification/threshold: This event indicates that the group’s balance reached or exceeded a threshold value; for example, when a purchase causes the balance to go over a 70% threshold.

By default, when this event occurs, the PCM_OP_ACT_POL_EVENT_NOTIFY policy opcode is called. You can add custom code to this policy opcode that processes upward threshold breaches, or you can enable this event to trigger a call to a balance monitoring opcode that processes upward threshold breaches (see "Editing the Event Notification List" in BRM Developer’s Guide).

- /event/notification/threshold_below: This event indicates that the group’s balance passed below a threshold value; for example, when the customer pays a bill. This event is generated only when the group’s balance is less than the threshold value.

By default, when this event occurs, the PCM_OP_ACT_POL_EVENT_NOTIFY policy opcode is called. To enable this event to trigger a call to a balance monitoring opcode that processes downward threshold breaches, see "Editing the Event Notification List" in BRM Developer’s Guide.

For more information about these events, see "About Notification Events for Balance Monitoring”.

To configure the event notification feature for balance monitoring:

1. If your system has multiple configuration files for event notification, merge them.

   See "Merging Event Notification Lists" in BRM Developer’s Guide.

2. Add information to your final event notification list about the appropriate opcodes to call when the /event/notification/threshold and /event/notification/threshold_below events occur.

   See "Editing the Event Notification List" in BRM Developer’s Guide.

---

Note:

- The default BRM_Home/sys/data/config/pin_notify file includes the following line, which means that the PCM_OP_ACT_POL_EVENT_NOTIFY (opcode number 301) policy opcode is called whenever an /event/notification/threshold event occurs:

  301 0 /event/notification/threshold

- The number of each balance monitoring opcode is in the balance monitor opcode header file, BRM_Home/include/ops/monitor.h.
Con FIGURING YOUR BRM SYSTEM FOR BALANCE MONITORING

1. Open the pin_rerate configuration file (BRM_Home/apps/pin_rerate/pin.conf) in a text editor.
2. Add the following entry and specify the delay time in seconds.

   - pin_rerate delay 300

3. Save and close the file.

Specifying a Wait Time before Rerating Events

You must configure pin_rerate to wait a specified amount of time before rerating events.

To configure a wait time:

1. Open the pin_rerate configuration file (BRM_Home/apps/pin_rerate/pin.conf) in a text editor.
2. Add the following entry and specify the delay time in seconds.

   - pin_rerate delay 300

3. Save and close the file.

Configuring pin_monitor_balance to Process Events in the Order Created

To process monitor balance impacts in chronological order, configure the pin_monitor_balance utility to run with only one thread. For more information, see "Using
Implementing Balance Monitoring in Custom Client Applications

To process monitor balance impacts in chronological order, perform the following tasks:

1. Open the `pin_monitor_balance` configuration file (`BRM_Home/apps/pin_monitor/pin.conf`) in a text editor.
2. Change the following entry to 1.
   - `pin_mta children 1`
3. Save and close the file.

Running `pin_monitor_balance` to Update Monitored Balances

To update monitored balances, perform the following:

1. Go to the `BRM_Home/apps/pin_monitor` directory.
2. Run the `pin_monitor_balance` utility:

   `pin_monitor_balance [-d] [-v]`

For more information, see `pin_monitor_balance`.

Specifying Whether Item Transfers Affect Balance Monitors

BRM updates balance monitor totals when you perform item adjustments, settlements, and disputes. By default, BRM also updates balance monitor totals when you perform item transfers. This can cause BRM to apply double the dispute, adjustment, or settlement amount to a balance monitor total when correcting misapplied charges. For example, when a customer disputes a misapplied charge, BRM reduces the balance monitor total when the charge is disputed and then again when the charge is transferred from the customer’s account to another account.

You can configure BRM to not apply balance impacts from item transfers to balance monitor totals by passing the optional `PIN_FLD_APPLY_MONITOR` input flist field to the `PCM_OP_BILL_ITEM_TRANSFER` opcode:

- When `PIN_FLD_APPLY_MONITOR` is set to 0, balance impacts from item transfers are not added to balance monitor totals.
- When `PIN_FLD_APPLY_MONITOR` is set to 1 or not passed, balance impacts from item transfers are added to balance monitor totals. This is the default.

For more information about adjustments, disputes, and settlements, see "Making Adjustments".

Implementing Balance Monitoring in Custom Client Applications

Before reading the following, you should be familiar with these concepts in BRM:

- Balance monitoring.
Implementing Balance Monitoring in Custom Client Applications

- Balance groups.
- Charge sharing.
- Discount sharing.
- Rating. For more information, see "How Rating Works" in BRM Setting Up Pricing and Rating.

About Implementing Balance Monitoring in Client Applications

To monitor balances, your Customer Service Representative (CSR) must be able to perform the following tasks:

- Specify the group of account and service balances to monitor.
- Update the group by adding or deleting members.
- View the balance monitors owned by an account.
- View the current balance of a balance monitor.
- Receive alerts when a balance monitor’s credit limit or threshold is crossed.

To implement balance monitoring in your client application, add the following functionality to your application by using the BRM opcodes:

- Create or update balance monitors.
  See "Creating and Maintaining Balance Monitor Objects".
- Display the balances for a monitor group or a list of monitors owned by an account or service.
  See "Displaying Balance Monitor Information in Client Applications".
- Send notifications when a credit limit or threshold is crossed.
  See "Updating Monitor Balances and Sending Credit Limit/Threshold Breach Notifications".

Your user interface must be designed to collect the information needed for creating or updating the relevant objects and pass the appropriate fields in the input flist to the opcodes.

Creating and Maintaining Balance Monitor Objects

To create or update a balance monitor, you must store the following balance monitor data in the BRM database:

1. The total rolled-up balance in a /balance_group/monitor object.
   You create only one such object for each balance monitor.
2. The list of monitor members in a /group/sharing/monitor object.
   You create only one such object for each balance monitor.
3. The list of monitor groups that each member belongs to in /ordered_balgrp objects.
   You must create or update one such object for each account and service in a balance monitor.

The method you use to create and update these objects depends on whether you are using Automated Monitor Setup (AMS), which is a group of opcodes that automate how BRM creates and maintains balance monitors.
To create or update balance monitors by using AMS, see "Using AMS to Create and Maintain Balance Monitors Automatically".

To create or update balance monitors without AMS, see "Using the Balance Monitor API to Create and Maintain Balance Monitors".

**Using AMS to Create and Maintain Balance Monitors Automatically**

For more information about AMS, see “About Using AMS to Manage Balance Monitors Automatically”.

To use AMS to create and update balance monitors automatically, you must configure your custom client application to call the following opcodes:

1. PCM_OP_CUST_SET_BAL_GRP: This opcode creates and updates the /balance_group/monitor object.
   - To create the object, see "Creating /balance_group/monitor Objects".
   - To modify the object, see "Modifying /balance_group/monitor Objects".

2. PCM_OP_SUBSCRIPTION_SHARING_GROUP_CREATE: This opcode creates the /group/sharing/monitor object.
   You must pass the monitor group type in the input flist. To create and maintain these objects, see "Creating, Modifying, or Deleting /group/sharing/monitor Objects".

When a /group/sharing/monitor object is created or modified, BRM uses the event notification feature to trigger calls to the appropriate AMS opcodes. These opcodes automatically add and remove members from /group/sharing/monitor objects and update each member’s /ordered_balgrp object based on the monitor type. See "Adding and Removing Balance Monitor Members Automatically".

**Important:** For subscription groups only, create a separate /ordered_balgrp object for any member service that uses a separate /billinfo object than its parent subscription service. You create /ordered_balgrp objects manually by using the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP opcode. See "Adding a Monitor Group to a Member’s /ordered_balgrp Object”.

**Using the Balance Monitor API to Create and Maintain Balance Monitors**

For more information, see "Managing Balance Monitors without AMS”.

To create or update balance monitors without AMS, you must configure your custom client application to call the following opcodes:

1. PCM_OP_CUST_SET_BAL_GRP: This opcode creates and updates the /balance_group/monitor object.
   - To create the object, see "Creating /balance_group/monitor Objects”.
   - To modify the object, see "Modifying /balance_group/monitor Objects”.

2. PCM_OP_SUBSCRIPTION_SHARING_GROUP_CREATE: This opcode creates the /group/sharing/monitor object.
   You must pass all group members in the input flist.
   - To create, modify, or delete the object, see "Creating, Modifying, or Deleting /group/sharing/monitor Objects”.

Balance Monitoring 13-17
To validate potential members before adding them to the object, see "Validating the Members of a Balance Monitor Group".

3. PCM_OP_SUBSCRIPTION_ORDERED_BALGRP: This opcode creates and updates /ordered_balgrp objects.

You must create or update the /ordered_balgrp object associated with each account and service that you want to include in a balance monitor. See "Adding a Monitor Group to a Member’s /ordered_balgrp Object".

For example, assume a balance monitor includes the following account hierarchy shown in Figure 13–1:

![Figure 13–1 Example Account Hierarchy with Balance Monitoring](image)

To include in the balance monitor the balance of all three accounts and their services, create or update the /ordered_balgrp object associated with each of the following:

- /account 1234
- /service/ip/cable 1234
- /account 1111
- /service/ip/cable 1111
- /service/ip/gprs 1111
- /account 2222
- /service/ip 2222

Creating /balance_group/monitor Objects

To create a /balance_group/monitor object, call the PCM_OP_CUST_SET_BAL_GRP opcode with the following information in the input flist:

- The owner of the balance monitor:
  - When the owner is an account, the POID of the /account object
  - When the owner is a service, the POIDs of the /account and /service objects
- Monitor name
- Credit limit
- Credit floor
Implementing Balance Monitoring in Custom Client Applications

- Credit thresholds

**Note:** You can also pass this information in the PIN_FLD_ACCTINFO array of the input flist to PCM_OP_CUST_COMMIT_CUSTOMER when creating an account. This opcode automatically calls PCM_OP_CUST_SET_BAL_GRP to create a /balance_group/monitor object when the relevant information is in the input flist and balance monitoring is enabled.

---

**Modifying /balance_group/monitor Objects**

To modify an existing /balance_group/monitor object, call the PCM_OP_CUST_SET_BAL_GRP opcode with the following information in the input flist:

- The owner of the balance monitor:
  - When the owner is an account, the POID of the /account object
  - When the owner is a service, the POIDs of the /account and /service objects
- The balance monitor to modify (POID of the /balance_group/monitor object)
- Monitor name
- Credit limit
- Credit floor
- Credit thresholds

**Note:** You can also pass this information in the PIN_FLD_ACCTINFO array of the input flist to PCM_OP_CUST_UPDATE_CUSTOMER when modifying an account. This opcode automatically calls PCM_OP_CUST_SET_BAL_GRP to modify the /balance_group/monitor object when the relevant information is in the input flist and balance monitoring is enabled.

---

**Deleting /balance_group/monitor Objects**

You cannot delete /balance_group/monitor objects. You can, however, deactivate the monitor group by deleting its associated /group/sharing/monitor object. To do so:

1. Run the pin_monitor_balance utility to update balances.
2. Delete the associated group/sharing/monitor object by calling the PCM_OP_SUBSCRIPTION_SHARING_GROUP_DELETE opcode.

This opcode removes the reference to the group/sharing/monitor object from corresponding /ordered_balgrp object. See "Creating, Modifying, or Deleting /group/sharing/monitor Objects".

---

**Creating, Modifying, or Deleting /group/sharing/monitor Objects**

To create or modify a /group/sharing/monitor object, you must collect the following information in your client application and pass it to the opcodes:

- Owner of the balance monitor:
  - When the owner is an account, the POID of the /account object
  - When the owner is a service, the POIDs of the /account and /service objects
Monitor group type: hierarchy, paying responsibility, or service level

List of members you want to add or delete

Balance monitor associated with the monitor group

You call the following opcodes from your client application:

To create a monitor group, call the PCM_OP_SUBSCRIPTION_SHARING_GROUP_CREATE opcode.

To modify a monitor group, call the PCM_OP_SUBSCRIPTION_SHARING_GROUP_MODIFY opcode.

To delete a monitor group, call the PCM_OP_SUBSCRIPTION_SHARING_GROUP_DELETE opcode.

These opcodes perform the following tasks to add, modify, or delete a monitor group object:

1. Verify balance monitoring is enabled.

2. (Create and modify opcodes only) Call the PCM_OP_SUBSCRIPTION_POL_PREP_MEMBERS policy opcode to validate the members.

   See "Validating the Members of a Balance Monitor Group".

3. Create, modify, or delete the /group/sharing/monitor object.

Changing the Owner of a Balance Monitor

To change the owner of a /group/sharing/monitor object, call the PCM_OP_SUBSCRIPTION_SHARING_GROUP_SET_PARENT opcode.

---

**Important:** This opcode can be used to change the owner of a balance monitor only if you are performing the change manually. Automated Monitor Setup (AMS) does not support changing owners.

---

If successful, this opcode returns the POID of the balance monitor that was modified and the event that was generated to record the owner change.

This opcode fails if the new owner that is passed is already a member of the resource sharing group.

To change the owner of a balance monitor, PCM_OP_SUBSCRIPTION_SHARING_GROUP_SET_PARENT does the following:

1. Verifies that:

   - AMS is not enabled. See "Enabling AMS in BRM".
   - The input flist contains the /balance_group object and /account object for the new balance monitor owner.
   - The /balance_group object belongs to the new owning account or service.

2. Sets the /group/sharing/monitor object’s owner to the new owner specified in the PIN_FLD_PARENT input field.

3. Creates an /event/group/sharing/monitor/modify event to record the owner change.
Validating the Members of a Balance Monitor Group
To validate the members of a balance monitor, call the PCM_OP_SUBSCRIPTION_POL_PREP_MEMBERS policy opcode. This opcode validates members before they are added or modified based on the monitor type, and returns a list of valid members to the calling opcode.

By default, this opcode validates the hierarchy, paying responsibility, and service level monitor types. You can create any type of monitor and validate it by implementing validation rules in this opcode.

The default implementation of the PCM_OP_SUBSCRIPTION_POL_PREP_MEMBERS opcode performs the following tasks:
1. Retrieves the monitor type.
2. For each member in the input flist, retrieves the account object.
3. Applies validation rules for the monitor type and returns a list of valid members for the group.

Adding a Monitor Group to a Member’s_/ordered_balgrp Object
The /ordered_balgrp object stores the list of sharing groups to which an account or service belongs. This object controls the order in which credits and discounts are applied and tracks the balance impacts for the balance groups.

Monitor groups are added to the PIN_FLD_ORDERED_BALGROUPS array of the /ordered_balgrp object. Unlike the charge sharing and discount sharing groups, the order of the monitor groups does not affect the balances. Therefore, the monitor groups are added to the end of the list.

To add a monitor group to the /ordered_balgrp object of an account or service, call the PCM_OP_SUBSCRIPTION_ORDERED_BALGRP opcode. This opcode performs the following tasks, depending on the value passed in the PIN_FLD_ACTION field in the input flist:
- Creates or deletes an /ordered_balgrp object.
- Modifies an /ordered_balgrp object by adding or deleting groups.
- Lists all sharing groups to which the account or service belongs.

This opcode performs the following tasks:
1. Verifies that the input flist has a /group/sharing/monitor object in the PIN_FLD_ORDERED_BALGROUPS array and that balance monitoring is enabled.

---

**Note:**
- When the input flist does not contain a /group/sharing/monitor object, the opcode continues with the tasks for processing the /group/sharing/charges and /group/sharing/discounts objects
- When balance monitoring is not enabled, the opcode returns an error.

2. Rearranges the PIN_FLD_ORDERED_BALGROUPS array to place the monitor groups at the end of the sequence.
3. Does one of the following tasks:
When the opcode is called to **create** the object, creates the `/ordered_balgrp` object for the specified account or service.

When the opcode is called to **modify** the object, modifies the `/ordered_balgrp` object by adding or deleting monitor group objects.

When the opcode is called to **delete** the object, deletes the `/ordered_balgrp` object.

4. Retrieves the current balance for the account or service `/balance_group`.

5. Generates an `/event/billing/monitor/update` or `/event/billing/monitor/delete` event.

6. Updates the balance of all associated `/balance_group/monitor` objects.

### Adding and Removing Balance Monitor Members Automatically

Use these AMS opcodes to manage balance monitors automatically.

**Important:** Do not call these opcodes directly. Call these opcodes only through the BRM event notification feature. See "Configuring Event Notification for Balance Monitoring".

- To automatically add members to a balance monitor when it is first created, use `PCM_OP_MONITOR_SETUP_MEMBERS`. See "Adding Members to Newly Created Balance Monitors Automatically".

- To automatically add or remove members from a balance monitor when an account hierarchy or subscription group changes, use these opcodes:
  - `PCM_OP_MONITOR_ACCOUNT_HIERARCHY`. See "Updating Hierarchy-Type Monitors Automatically".
  - `PCM_OP_MONITOR_BILLING_HIERARCHY`. See "Updating Paying Responsibility-Type Monitors Automatically".
  - `PCM_OP_MONITOR_SERVICE_HIERARCHY`. See "Updating Subscription-Type Monitors Automatically".
  - `PCM_OP_MONITOR_HIERARCHY_CLEANUP`. See "Removing Members from Hierarchy- and Paying Responsibility-Type Monitors".

### Adding Members to Newly Created Balance Monitors Automatically

Use the `PCM_OP_MONITOR_SETUP_MEMBERS` opcode to automatically add members to a balance monitor when it is first created. This opcode is a wrapper opcode that, according to monitor group type, calls other standard MONITOR opcodes to:

- Find and add members to the monitor group (`/group/sharing/monitor` object).

- Update each member’s ordered balance group list (`/ordered_balgrp` object).

- Add each member’s current balance to the group balance (`/balance_group/monitor` object).

This opcode is triggered by the `/event/group/sharing/monitor/create` event.
Implementing Balance Monitoring in Custom Client Applications

**Important:** Do not call this opcode directly. This opcode should be called through the event notification feature only. See "Configuring Event Notification for Balance Monitoring".

This opcode retrieves the monitor type and monitor owner, and then calls one of these opcodes:

- **For hierarchy group types (H_CE):** Calls PCM_OP_MONITOR_PROCESS_HIERARCHY_MONITORS. This opcode adds the following members to the balance monitor:
  - The parent account and its services
  - All nonpaying child accounts and their services
  - All paying child accounts and their services

- **For paying responsibility group types (PR_CE):** Calls PCM_OP_MONITOR_PROCESS_BILLING_MONITORS. This opcode adds the following members to the balance monitor:
  - The parent account and its services
  - All nonpaying child accounts and their services

- **For service group types (SUB_CE):** Calls PCM_OP_MONITOR_PROCESS_SERVICE_MONITORS. This opcode adds the following members to the balance monitor:
  - The parent subscription service
  - All member services

For more information about these opcodes, see "Balance Monitoring FM Standard Opcodes" in BRM Developer’s Reference.

**Adding Members to Hierarchy-Type Monitors**

Use the PCM_OP_MONITOR_PROCESS_HIERARCHY_MONITORS opcode to add members to hierarchy-type balance monitors (see BRM Developer’s Reference). This opcode finds and adds the following members to the balance monitor:

- The parent account and its subscriptions
- All paying child accounts and their subscriptions
- All nonpaying child accounts and their subscriptions

This opcode is called directly by the PCM_OP_MONITOR_SETUP_MEMBERS wrapper opcode.

PCM_OP_MONITOR_PROCESS_HIERARCHY_MONITORS performs the following tasks:

1. Finds all services that are associated with the parent account.
2. Finds in the hierarchy all paying and nonpaying child accounts that belong to the parent account.
3. Finds all services that are associated with each child account.
4. Adds the following members to the balance monitor:
   - The parent account and its services.
Implementing Balance Monitoring in Custom Client Applications

- All child accounts and their services.
  See "Creating, Modifying, or Deleting /group/sharing/monitor Objects".

5. Updates each account’s and service’s /ordered_balgrp object.
  See "Adding a Monitor Group to a Member’s /ordered_balgrp Object”.

Adding Members to Payment Responsibility-Type Monitors
Use the PCM_OP_MONITOR_PROCESS_BILLING_MONITORS opcode to add members to payment responsibility-type balance monitors. This opcode finds and adds the following members to the balance monitor:
- The parent account and its subscriptions
- All nonpaying child accounts and their subscriptions
This opcode is called directly by the PCM_OP_MONITOR_SETUP_MEMBERS wrapper opcode.

PCM_OP_MONITOR_PROCESS_BILLING_MONITORS performs the following tasks:
1. Retrieves the parent’s bill unit (billinfo) from either the input flist or searches....
2. Searches for all the bill_info objects that have PIN_FLD_PARENT_BILLINFO_OBJ as the current billinfo.
3. Searches for the owning account and all services for each billinfo found.
4. Adds the following members to the balance monitor:
   - The parent account and its services
   - All nonpaying child accounts and their services.
   See "Creating, Modifying, or Deleting /group/sharing/monitor Objects”.
5. Updates each account’s and service’s /ordered_balgrp object.
   See "Adding a Monitor Group to a Member’s /ordered_balgrp Object”.

Updating Hierarchy-Type Monitors Automatically
Use the PCM_OP_MONITOR_ACCOUNT_HIERARCHY opcode to add members to hierarchy-type balance monitors when an account hierarchy changes (see BRM Developer’s Reference). For example, this opcode adds members to balance monitors when any of the following actions affect an account hierarchy:
- An account is added to the account hierarchy
- A child account adds a service
- The association between a balance group and a bill unit changes
- A line is transferred to an account member
This opcode is triggered by the following events:
- /event/group/member
- /event/notification/service/pre_purchase
- /event/audit/subscription/transfer
Implementing Balance Monitoring in Custom Client Applications

**Important:** Do not call this opcode directly. This opcode should be called through the event notification feature only. See "Configuring Event Notification for Balance Monitoring”.

---

PCM_OP_MONITOR_ACCOUNT_HIERARCHY finds all parent accounts at a higher level in the hierarchy than the specified account or service. For each parent account, the opcode performs the following tasks:

1. Finds all hierarchy-type monitors associated with the parent account.
2. Adds the account or service to the hierarchy-type monitor groups (/group/sharing/monitor objects).
3. Adds the monitor group to the account’s or service’s ordered balance group list (/ordered_balgrp object).
4. Adds the account or service balance to the monitored balance (/balance_group/monitor objects).

---

**Important:** Do not call this opcode directly. This opcode should be called through the event notification feature only. See "Configuring Event Notification for Balance Monitoring”.

---

PCM_OP_MONITOR_BILLING_HIERARCHY finds all parent accounts at a higher level in the hierarchy than the specified account or service. For each parent account, the opcode performs the following tasks:

1. Finds all paying responsibility-type monitors associated with the parent account.
2. Adds the account or service to the paying responsibility-type monitor groups (/group/sharing/monitor objects).
3. Adds the monitor group to the account’s or service’s ordered balance group list (/ordered_balgrp object).
4. Adds the account or service balance to the monitored balance (/balance_group/monitor objects).

---

**Important:** Do not call this opcode directly. This opcode should be called through the event notification feature only. See "Configuring Event Notification for Balance Monitoring”.

---

Updating Paying Responsibility-Type Monitors Automatically
Use the PCM_OP_MONITOR_BILLING_HIERARCHY opcode to add members to paying responsibility-type monitors when changes occur in an account hierarchy. For example, this opcode adds members to balance monitors when any of the following actions affect an account hierarchy:

- A nonpaying child account is added to the account hierarchy
- A nonpaying child account adds a service
- A child account changes from a paying to a nonpaying payment type
- The association between a balance group and a bill unit changes
- A line is transferred to a nonpaying child account

This opcode is triggered by the following events:

- /event/notification/service/pre_purchase
- /event/customer/billinfo/modify
- /event/notification/bal_grp/modify

---

**Important:** Do not call this opcode directly. This opcode should be called through the event notification feature only. See "Configuring Event Notification for Balance Monitoring”.

---

PCM_OP_MONITOR_BILLING_HIERARCHY finds all parent accounts at a higher level in the hierarchy than the specified account or service. For each parent account, the opcode performs the following tasks:

1. Finds all paying responsibility-type monitors associated with the parent account.
2. Adds the account or service to the paying responsibility-type monitor groups (/group/sharing/monitor objects).
3. Adds the monitor group to the account’s or service’s ordered balance group list (/ordered_balgrp object).
4. Adds the account or service balance to the balance monitors (/balance_group/monitor objects).

**Updating Subscription-Type Monitors Automatically**
Use the PCM_OP_MONITOR_SERVICE_HIERARCHY opcode to add members to subscription-type monitors when a subscription group changes. For example, when you add a member service to a subscription, this opcode adds the service to any associated balance monitors.

This opcode is triggered by the /event/notification/service/pre_purchase notification event.

---

**Important:** Do not call this opcode directly. This opcode should be called through the event notification feature only. See "Configuring Event Notification for Balance Monitoring".

---

PCM_OP_MONITOR_SERVICE_HIERARCHY finds all parent subscription services at a higher level in the group than the specified service. For each parent subscription service, the opcode performs the following tasks:

1. Finds all subscription-type monitors associated with the parent subscription service.
2. Adds the service to the subscription-type monitor groups (/group/sharing/monitor objects).
3. Determines whether the added service is a subscription service:
   - If it is, proceeds to the next step.
   - If it is not, adds the monitor groups to the service’s /ordered_balgrp object.
4. Adds the service balance to the balance monitors (/balance_group/monitor objects).

**Removing Members from Hierarchy- and Paying Responsibility-Type Monitors**
Use the PCM_OP_MONITOR_HIERARCHY_CLEANUP opcode to remove members from hierarchy-type or paying responsibility-type monitors when an account hierarchy changes. For example, this opcode is called when any of the following actions affect an account hierarchy:

- An account is removed from the hierarchy
- A child account changes from a nonpaying to a paying payment type
- The association between a balance group and a bill unit changes
- A line is removed from a child account

This opcode is triggered by the following events:

- /event/customer/billINFO/modify
- /event/group/member
- /event/notification/bal_grp/modify
Implementing Balance Monitoring in Custom Client Applications

Displaying Balance Monitor Information in Client Applications

You can retrieve and display the following information in your client application:

- The balances for a monitor group
- A list of monitor groups owned by an account or service

Retrieving the Balances for a Monitor Group

To retrieve the total rolled-up balance for a monitor group, call the PCM_OP_BAL_GET_MONITOR_BAL opcode. The opcode retrieves either the balance monitor’s current balance or the balance total for a specified time period.

The opcode returns one of the following, depending on whether you pass the PIN_FLD_DATE_BALANCES array in the input flist:

- If you pass the array with a start and end date, the opcode returns the total amount generated by members between those days.
- If you do not pass the array, the opcode returns the balance monitor’s current rolled-up balance.

This opcode performs the following tasks to retrieve the balances:

1. Retrieves the POID of the \balance_group/monitor object and, optionally, the dates for which the balances are requested.
2. Reads the \balance_group/monitor object and retrieves the balance information.
3. When the input flist contains dates, retrieves the correct bucket for the specified dates from the sub-balances.
4. Returns the balance for the specified monitor group.

Retrieving the Balance Monitors Owned by an Account or Service

To retrieve a list of balance monitors owned by an account or service, call the PCM_OP_BAL_GET_ACCT_MONITORS opcode.

This opcode takes as input the POID of the \account or \service object and performs the following tasks:

1. Searches for all the \group/sharing/monitor objects associated with the account or service specified in the input flist.
2. For each \group/sharing/monitor object associated with the account or service, retrieves the \balance_group/monitor objects.
3. Returns a list of monitor groups owned by the specified account or service.

Important: Do not call this opcode directly. This opcode should be called through the event notification feature only. See "Configuring Event Notification for Balance Monitoring".

Note: The end date is exclusive. For example, if you enter January 1 as the start date and January 5 as the end date, the opcode sums all balance impacts generated on January 1, 2, 3, and 4.
Updating Monitor Balances and Sending Credit Limit/Threshold Breach Notifications

When a credit limit or threshold is reached, BRM generates a notification event containing information about the affected account and the reason for the breach. (See "About Using Event Notification to Alert Customers").

When you run the `pin_monitor_balance` utility to update monitored balances with the current monitor impacts, it calls the `PCM_OP_BAL_APPLY_MONITOR_IMPACTS` opcode. This opcode updates the `/balance_group/monitor` objects and also compares the monitored balance with the credit limit and threshold values stored in the `/config/credit_profile` object. It generates notification events when the credit limits or thresholds are crossed.

The `PCM_OP_BAL_APPLY_MONITOR_IMPACTS` opcode generates two kinds of notification events:

- `/event/notification/threshold` when the balance crosses above the credit threshold
- `/event/notification/threshold_below` when the balance falls below the credit threshold

**Important:** These are notification events only and are not recorded in the database. See "About Notification Events" in BRM Developer’s Guide.

Table 13–2 lists the fields in the notification events.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_RESOURCE_ID</td>
<td>The resource ID.</td>
</tr>
<tr>
<td>PIN_FLD_AMOUNT</td>
<td>The balance impact for this event.</td>
</tr>
<tr>
<td>PIN_FLD_BAL_GRP_OBJ</td>
<td>The POID of the <code>/balance_group/monitor</code> object.</td>
</tr>
<tr>
<td>PIN_FLD_PERCENT</td>
<td>The percentage amount crossed.</td>
</tr>
<tr>
<td>PIN_FLD_SOURCE_OBJ</td>
<td>Source of the breach, for example, the POID of the event that caused the breach.</td>
</tr>
<tr>
<td>PIN_FLD_ALERT_TYPE</td>
<td>The alert type: credit limit or threshold percentage.</td>
</tr>
<tr>
<td>PIN_FLD_MONITOR_TYPE</td>
<td>Type of monitor:</td>
</tr>
<tr>
<td></td>
<td>- Paying responsibility credit exposure</td>
</tr>
<tr>
<td></td>
<td>- Hierarchy credit exposure</td>
</tr>
<tr>
<td></td>
<td>- Service level</td>
</tr>
<tr>
<td></td>
<td>- A type that you have defined.</td>
</tr>
</tbody>
</table>
You can use the information in the events to notify the CSR through a client application or the customer through email, for example, by writing a custom application.

For more information about using event notification, see the following topics:

- "Using Event Notification" in BRM Developer’s Guide
- About Using Event Notification to Alert Customers
- Configuring Event Notification for Balance Monitoring

### Example of Credit Threshold Notification Event Generation

Consider a balance monitor for an account or service with:

- A credit floor of $0
- A credit limit of $100
- Thresholds set at 25%, 75%, and 90%
- A current balance of $50

When a new event comes in with a balance of $26, the new balance is set to $76, and a notification event with the following information is generated:

```
PIN_FLD_ALERT_TYPE    Threshold
PIN_FLD_REASON        Upward breach
PIN_FLD_CREDIT_THRESHOLDS  75%
```

## Table 13–2 (Cont.) Notification Event Fields for Monitoring

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PIN_FLD_REASON</td>
<td>The reason for the breach:</td>
</tr>
<tr>
<td></td>
<td>- Upward breach</td>
</tr>
<tr>
<td></td>
<td>- Downward breach</td>
</tr>
<tr>
<td></td>
<td>- Upward reset</td>
</tr>
<tr>
<td></td>
<td>- Downward reset</td>
</tr>
<tr>
<td></td>
<td>- Unknown reason</td>
</tr>
<tr>
<td>PIN_FLD_THRESHOLD</td>
<td>Tracks when the balance of a monitor group crosses a 5% boundary. For example, it tracks whether the current balance is 5%, 10%, or 15% of the credit limit.</td>
</tr>
<tr>
<td>PIN_FLD_CREDIT_THRESHOLDS</td>
<td>An array of credit thresholds and limits breached by this event.</td>
</tr>
</tbody>
</table>

**Important:** The opcode generates only one event if multiple thresholds are crossed. However, if multiple monitors cross the thresholds because of a single balance impact, the utility generates an event for each monitor.
This chapter provides reference information for Oracle Communications Billing and Revenue Management (BRM) accounts receivable utilities.

Topics in this document:

- `pin_apply_bulk_adjustment`
- `pin_mass_refund`
- `pin_refund`
pin_apply_bulk_adjustment

Use this BRM command-line utility to apply an adjustment across a broad range of accounts, such as all accounts charged a regular day-time rate when they should have been charged a holiday rate. See “Adjusting Multiple Accounts Simultaneously”

Location

BRM_home/bin

Syntax

`pin_apply_bulk_adjustment [-verbose file_name.log] [ -test ] [ -h ] -f input_file.csv`

Parameters

- **-verbose**
  Displays information about successful or failed processing as the utility runs.

- **-h**
  Displays the utility’s syntax and parameters.

- **-f input_file.csv**
  Specifies the name and location of the file that specifies how BRM will apply the bulk adjustment; for example, C:\bulkadj\bulk_run_1.csv.

Results

Each account adjustment initiated during a bulk adjustment is handled as a separate transaction. Therefore, if any single adjustment in the bulk adjustment fails, BRM does not need to roll back the entire bulk adjustment. Rather, the adjustment remains in place for all accounts that had successful adjustments, and the utility identifies the accounts that failed to adjust so you can correct the problem. The utility provides this feedback in a log file (default.pinlog) that it generates in its start directory or in a directory specified by the configuration file.
pin_monitor_balance

Use this utility to update monitored balances and to check whether the balances have crossed thresholds.

For more information on using the *pin_monitor_balance* utility, see "Running *pin_monitor_balance* to Update Monitored Balances"

---

**Note:** To connect to the BRM database, this utility needs a configuration file in the directory from which you run it. The *pin.conf* file for this utility is in `BRM_Home/apps/pin_monitor`. See "Creating Configuration Files for BRM Utilities" in *BRM System Administrator’s Guide*.

---

**Location**

`BRM_Home/bin`

**Syntax**

```
pin_monitor_balance [-d] [-v] [-t] [-h]
```

**Parameters**

- `-d`
  Prints error logs for debugging.

- `-v`
  Displays detailed information on status and error messages as the utility updates balances and generates notification events.

- `-t`
  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 `-v` option.

- `-h`
  Displays the syntax and parameters for this utility.

**Results**

This utility generates notification events when a balance crosses a credit threshold. It logs errors in the log file, which is either in the directory from which the utility was started or in a directory specified in the configuration file.
pin_mass_refund

This utility creates a refund for accounts that have a credit balance. See "Giving Refunds to Customers".

**Important:** For multischema systems, you must run the utility separately against each database schema in your system. See "Running Non-MTA Utilities in Multischema Systems" in *BRM System Administrator’s Guide*.

### Location

`BRM_home/bin`

### Syntax

```
pin_mass_refund -active | close | inactive
 [-pay_type payment_method]
 [-test] [-verbose file_name.log] [-test] [-help]
```

### Parameters

- **-active | close | inactive**
  Specifies the status of the accounts for which to create refund items.

- **-pay_type payment_method**
  Specifies the payment method, for example:

  - 10003 for credit card
  - 10005 for direct debit
  - 10018 for SEPA

- **-test**
  Finds the bill units (/billinfo objects) that meet the criteria but does not create any refund objects.

- **-verbose**
  Displays information about successful or failed processing as the utility runs.

- **-help**
  Displays 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.
pin_refund

This utility finds accounts that have refund items and makes BRM-initiated refund payments to customers. See "Giving Refunds to Customers".

When you use multiple payment processors, you run this utility for each one. See "Using More Than One Payment Processor" in BRM Configuring and Collecting Payments.

---

**Important:** For multischema systems, you must run the utility separately against each database schema in your system. See "Running Non-MTA Utilities in Multischema Systems" in BRM System Administrator’s Guide.

---

Location

BRM_home/bin

Syntax

pin_refund  -pay_type  payment_type_indicator  
[ -vendor  payment_processor_name  
[ -active  | close  | inactive  
[ -verbose  file_name.log  ]  [-test]  [-help]

Parameters

**-pay_type  payment_type_indicator**
Specifications the payment type, for example:

- 10003 for credit card
- 10005 for direct debit
- 10018 for SEPA

**-vendor  payment_processor_name**
Specifications the credit card processor or automated clearing house (ACH) to use for validating credit cards, debit cards, and direct debit transactions.

This parameter is not applicable for the SEPA payment type.

For information on configuring payment processor information, see BRM Configuring and Collecting Payments.

**-active | close | inactive**
Specifies the status of the accounts to give refunds to.

**-verbose**
Displays information about successful or failed processing as the utility runs.

**-test**
Returns the number of refund items that would be created, but does not create any refund items.

**-help**
Displays 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_refund utility logs error information in the pin_billd.pinlog file.