# Oracle® Communications Billing and Revenue Management

## Rerating Pipeline Events

Release 15.1
F93244-01
April 2025

Oracle Communications Billing and Revenue Management Rerating Pipeline Events, Release 15.1

F93244-01

# Contents

## 2 About Real-Time Rerating Pipelines



## 3 About Rerating Pipeline-Rated Events



## 4 Using Event Extraction Manager

## 5 Configuring Rerating in Pipeline Manager



## 6 About Comprehensive Rerating Using pin_rerate

## 7 Configuring Comprehensive Rerating

## 8 Using the pin_rerate Utility

## 9 Rerating Utilities

# Preface

This book describes how to configure and run the event rerating process for pipeline-rated events in Oracle Communications Billing and Revenue Management.

## Audience

This guide is intended for system administrators.

## Documentation Accessibility

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

**Access to Oracle Support**

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

## Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

# About Rerating Events

Learn about the Oracle Communications Billing and Revenue Management (BRM) rerating process.

Topics in this document:



## About Rerating

BRM rerating is the process you use to rerate events that were originally rated in batch by Pipeline Manager or in real time. You might rerate events for several reasons:

- To rerate a group of accounts after changing a charge offer
- To rerate an account when a customer service representative (CSR) backdates a subscriber's purchase or cancellation
- To back out events when call detail records (CDRs) contain field errors
- To rerate events when the rating conditions change during the session

> **Note:**
>
> – Member, child, and subordinate accounts in charge sharing, discount sharing, and account sponsorship and hierarchy groups are not automatically rerated when the parent or sponsoring account is rerated. For more information, see "About Rerating Events for Account Sharing Groups".
>
> – You can rerate remittance accounts, but rerating subscriber accounts does not automatically rerate the remittance accounts. Events already included in remittance processing are not rerated.

The balance impact of rerating is the difference between the original rated event and the rerated event. This difference is applied to the account balance in the current billing cycle. For more information, see "How BRM Applies the Balance Impacts of Rerating".

# About the Rerating Process

The rerating process consists of the following steps:

1. **Extracting events**. BRM extracts the events to rerate for an account or a group of accounts from the BRM database.
2. **Restoring balances**. BRM backs out the original event balance impacts and restores the account's discount and aggregation balances to their original states before rating the events.
3. **Calculating new balances**. BRM rerates the events based on new pricing information and calculates the new discount and aggregation balances.
4. **Recording the event**. BRM loads the events with the new balance impacts into the BRM database, updating all relevant account data.

# About Automatic Allocation from Rerating

By default, BRM creates unallocated items for any adjustment items that are created as a result of rerating.

## Corrective Billing and Automatic Allocation of Rerating Adjustments

If your corrective bills must contain the aggregation and allocation of automatic adjustments to the items on the original bill, you must enable the **AllocateReratingAdjustments** business parameter before you rerate the original bills. When the **AllocateReratingAdjustments** business parameter is enabled, adjustment details by original item by original bill are reported on the next bill for regular billing also.

To ensure that the corrective billing process includes or excludes these adjustments, check the setting for the **AllocateReratingAdjustments** business parameter *before* you run the rerating process.

If BRM has rerated a **/bill** object without allocating automatic adjustments (from the rerating) to the original bills, the corrective bill for that **/bill** object will not include the item adjustments generated by that rerating.

> **Note:**
>
> For such a **/bill** object, if you set the **AllocateReratingAdjustments** business parameter to **enabled** and rerun the rerating process for the bill, BRM does not allocate the adjustments.
>
> You must manually allocate the rerated items *before* you generate the corrective bill for such a **/bill** object.

If you enable BRM to automatically allocate the item adjustments generated by the rerating process to the original bill, BRM allocates adjustments to the bill items being corrected in the following manner:

- For open item accounting, the adjustment items are allocated to each bill that was corrected. Allocation is made to each of the original bills that included events or items that were corrected by these adjustments.
- For balance forward accounting, corrections are posted to the last bill only. Therefore, the rerating allocation is made to the final bill only. This final bill carries over the balances from all prior bill periods.

## Enabling Automatic Allocation of Rerating Adjustments

To enable automatic allocation of rerating adjustments:

1. Go to the *BRM_home*/**sys/data/config** directory, where *BRM_home* is the directory in which BRM is installed.
2. Run the following command, which creates an editable XML file from the **rerate** instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsRerate bus_params_rerate.xml
   ```

   This command creates the XML file named **bus_params_rerate.xml.out** in your working directory. To place this file in a different directory, specify the path as part of the file name.
3. Open the **bus_params_rerate.xml.out** file.
4. Search for the following line:

   ```
   <AllocateReratingAdjustments>disabled</AllocateReratingAdjustments>
   ```
5. Change **disabled** to **enabled**.
6. Save the file as **bus_params_rerate.xml**.
7. Go to the *BRM_home*/**sys/data/config** directory, which includes support files used by the **pin_bus_params** utility.
8. Run the following command, which loads this change into the appropriate **/config/business_params** object:

   ```
   pin_bus_params PathToWorkingDirectory/bus_params_rerate.xml
   ```

   where *PathToWorkingDirectory* is the directory in which **bus_params_rerate.xml** resides.

   > **Caution:**
   >
   > BRM uses the XML in this file to overwrite the existing **rerate** 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 **rerate** configuration.

9. Read the object with the **testnap** utility or Object Browser to verify that all fields are correct.
10. Stop and restart the Connection Manager (CM).

# About Deferred Taxes During Rerating

By default, BRM does not compute deferred tax during rerating.

# Enabling Calculation of Deferred Taxes During Rerating

> **Note:**
>
> When deferred taxation is enabled for rerating:
>
> - You cannot use selective rerating.
> - You cannot rerate events created by Bill Now.
> - You cannot rerate events that use multiple tax suppliers.

By default, BRM calculates taxes on any deferred taxable amount in the rerated events during the subsequent bill run. The rerated tax appears on the invoice for the subsequent bill run. Calculating taxes during rerating using the **ApplyDeferredTaxDuringRerating** business parameter enables corrected invoices to show the corrected tax amount. When combined with corrective invoicing, rerated events will occur and their tax amounts will appear on the invoice for the original billing period.

To enable calculation of deferred taxes during rerating:

1. Go to the *BRM_home*/**sys/data/config** directory.
2. Run the following command, which creates an editable XML file from the **rerate** instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsRerate bus_params_rerate.xml
   ```

   This command creates the XML file named **bus_params_rerate.xml.out** in your working directory. To place this file in a different directory, specify the path as part of the file name.
3. Open the **bus_params_rerate.xml.out** file.
4. Search for the following line:

   ```
   <ApplyDeferredTaxDuringRerating>disabled</ApplyDeferredTaxDuringRerating>
   ```
5. Change **disabled** to **enabled**.
6. Save the file as **bus_params_rerate.xml**.
7. Go to the *BRM_home*/**sys/data/config** directory, which includes support files used by the **pin_bus_params** utility.
8. Run the following command, which loads this change into the appropriate **/config/business_params** object:

   ```
   pin_bus_params PathToWorkingDirectory/bus_params_rerate.xml
   ```

   where *PathToWorkingDirectory* is the directory in which **bus_params_rerate.xml** resides.

> **Caution:**
>
> BRM uses the XML in this file to overwrite the existing **rerate** 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 rerate configuration.

9. Read the object with the **testnap** utility or Object Browser to verify that all fields are correct.
10. Stop and restart the CM.

# About Rerating Events by Using the Rates Applied when the Rating Conditions Change During the Session

By default, events are rerated in order of occurrence based on the event end time. Rerating the events using the **OfferEligibilitySelectionMode** business parameter enables to rerate the events in order of event start time when the rating conditions change during the session. For example, if during a call session, the subscriber adds the called number of that session to a Friends and Family list, BRM applies the Friends and Family discount for the session from the time the called number is added to the Friends and Family list. When the call is rerated, the actual rate is applied from the start time of the call until the called number is added to the Friends and Family list, and the discount is applied for the remaining session.

## Enabling Rerating when the Rating Conditions Change During the Session

To enable rerating when the rating conditions change during the session:

1. Go to the *BRM_home***/sys/data/config** directory, where *BRM_home* is the directory in which BRM is installed.
2. Run the following command, which creates an editable XML file from the **rerate** instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsRerate bus_params_rerate.xml
   ```

   This command creates the XML file named **bus_params_rerate.xml.out** in your working directory. To place this file in a different directory, specify the path as part of the file name.
3. Open the **bus_params_rerate.xml.out** file.
4. Search for the following line:

   ```
   <OfferEligibilitySelectionMode>endtime</OfferEligibilitySelectionMode>
   ```

5. Change **endtime** to **timeperiod**.
6. Save the file as **bus_params_rerate.xml**.
7. Go to the *BRM_home***/sys/data/config** directory, which includes support files used by the **pin_bus_params** utility.
8. Run the following command, which loads this change into the appropriate **/config/business_params** object:

   ```
   pin_bus_params PathToWorkingDirectory/bus_params_rerate.xml
   ```

   where *PathToWorkingDirectory* is the directory in which **bus_params_rerate.xml** resides.

> **Caution:**
>
> BRM uses the XML in this file to overwrite the existing **rerate** 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 **rerate** configuration.

9. Read the object with the **testnap** utility or Object Browser to verify that all fields are correct.
10. Stop and restart the CM.

# Understanding the BRM Rerating Features

Using BRM, you can rerate events in the following ways:

- **By using Pipeline Manager to rerate pipeline-rated events only**. This method is used for rerating events originally rated by Pipeline Manager.

  With this method, you extract the events for rerating from the BRM database by using the **pin_event_extract** utility. You then rerate the events in a batch rerating pipeline.

  Events are rerated and recorded using a two-step process: Events are rerated by the pipeline and then recorded in the BRM database when they are loaded by Rated Event (RE) Loader. As a result, pipeline event balance impacts are applied when they are recorded in the BRM database (event creation time).

  Because an account can also have real-time event balance impacts that are recorded in real time (event end time), there is a possibility that the pipeline-rated and real-time rated event balance impacts are applied out of sequence, which might affect how BRM bills for the usage.

  > **Note:**
  >
  > When using Pipeline Manager to rerate pipeline-rated events, you must determine whether this method of rerating is sufficient to provide consistent results. Accounts with pipeline-rated usage may also have real-time events, such as purchase events, cancel events, or cycle fee events that can impact the rating of usage events.

- **By using the pin_rerate utility to rerate pipeline-rated and real-time rated events**. This method can be used for rerating events originally rated in real time and events originally rated in batch by Pipeline Manager.

  With this method, you select events for rerating and rerate the events by running the **pin_rerate** utility. Pipeline-rated events are rerated in a real-time rerating pipeline and real-time rated events are rerated by real-time rating opcodes.

  Pipeline-rated and real-time rated events are rerated and recorded in the same process. Events are rerated in order, based on the time they occurred (the event end time). This results in pipeline-rated *and* real-time-rated event balance impacts being applied in the correct sequence. For more information, see "About Rerating Pipeline and Real-Time Events Concurrently".

  If you do not use Pipeline Manager for batch rating, you can also use **pin_rerate** to rerate only real-time-rated events.

> **Note:**
>
> Using **pin_rerate** to rerate events can degrade system performance, depending on the system load.
>
> When you have a high volume of events to rerate (for example, more than 100,000), you might be able to use Pipeline Manager to rerate pipeline-rated events and use **pin_rerate** to rerate real-time rated events to reduce the system load. You can do this if the accounts being rerated have only real-time-rated events or have only pipeline-rated events. If accounts have both real-time rated and pipeline-rated events, you should use **pin_rerate** to ensure consistent results.

## About Rerating Pipeline and Real-Time Events Concurrently

When you use **pin_rerate** to rerate pipeline-rated and real-time-rated events concurrently, by default, events are rerated in order of occurrence based on the event end time and recorded in the BRM database as they are processed. This process applies the pipeline and real-time event balance impacts in the sequence that the original real-time usage occurred.

For example, $R_1$ and $R_2$ are real-time-rated events and $P_1$ and $P_2$ are pipeline-rated events. The sequence in which these events were generated in real time was $R_1$, $P_1$, $P_2$, $R_2$. However, the order in which they were recorded in the BRM database was $R_1$, $R_2$, $P_1$, $P_2$. When these events are rerated by **pin_rerate**, the events are rerated and recorded in the original sequence in which they occurred ($R_1$, $P_1$, $P_2$, $R_2$).

You can also use **pin_rerate** parameters to rerate the events in the order they were originally *recorded* ($R_1$, $R_2$, $P_1$, $P_2$).

## What You Can Do with Rerating

When using Pipeline Manager to rerate pipeline-rated events only, you can do the following:

- Select accounts and events for rerating based on various criteria, such as:
  - Account number
  - Accounts that have events related to specific bundles, charge offers, discount offers, or specific event and service types
  - Events associated with specific charge offers, discounts, subscription services, or bill units
- Perform back-out-only rerating, which backs out the balance impacts of rating without reapplying new balance impacts.

When using **pin_rerate** to rerate pipeline-rated and real-time-rated events, you can do the following:

- Select accounts and events for rerating based on various criteria such as account number, bundles, charge offers, discounts, event type, service type, bill unit and so on.

  Rerating a bill unit rerates all usage events, cycle events, billing-time discounts, folds, and rollover events associated with the bill unit.
- Perform back-out-only rerating, which backs out the balance impacts of rating without reapplying new balance impacts.

- Rerate real-time-rated and pipeline-batch-rated events concurrently or, if you do not use Pipeline Manager for batch rating, rerate only real-time-rated events.
- Assign a rerate reason code to accounts selected for rerating. This enables you to rerate only accounts matching the reason for rerating.
- Set up automatic rerating, which automatically selects certain events for rerating so that you do not need to specify the accounts and events when you run the **pin_rerate** utility. You can set up the following automatic rerating features:
  - Automatic rerating for backdated events, rate changes, and rollover corrections
  - Out-of-order rerating for pipeline-rated events that are rated out of order (non-chronologically)
  - Trigger-dependent rerating for events based on custom rerating triggers that you configure. For example, you can automatically select events for rerating that are rated for a charge offer that has been canceled.
- Create custom **pin_rerate** parameters, which enables you to select events based on any event criteria.

## About the Rerating Pipelines

Pipeline Manager rerates only the events that it previously rated. Pipeline Manager has two rerating pipelines. The one you use depends on whether you rerate events by using Pipeline Manager or by using **pin_rerate**:

- **The batch rerating pipeline**. This pipeline is used when rerating only pipeline-rated events by using Pipeline Manager. It receives and rerates events in batches. The rerated events are loaded into the BRM database in batches by RE Loader.
- **The real-time rerating pipeline**. This pipeline is used when rerating pipeline-rated events by using **pin_rerate**. It rerates events individually that it receives from the CM. The rerated events are recorded into the BRM database as they are processed.

# How BRM Applies the Balance Impacts of Rerating

BRM rerating handles both billed and unbilled events and rerating events across billing cycles. When events include noncurrency balance impacts, BRM rerating properly redistributes noncurrency balance elements and charges or credits the account accordingly. Financial impacts to accounts receivable (A/R), general ledger (G/L), and taxation are taken into account when adjustments are made as a result of rerating.

When an event is rerated, BRM backs out the event from the BRM database by creating an adjustment event that fully negates the original balance impacts. This adjustment event can be either a shadow adjustment event or a regular adjustment event:

- If the event is unbilled, a shadow adjustment event is created. See "About Rerating Unbilled Events".
- If the event has already been billed or posted to the G/L, a regular adjustment event is created. See "About Rerating Billed and Posted Events".

For rerated usage events, the balance impacts of rerating are applied by the adjustment event. For rerated cycle events, a new cycle event is created that applies the balance impacts of rerating. Cycle events include cycle fees, folds, rollovers, and cycle discounts.

When the total rerating adjustment is zero, BRM does not generate a rerating adjustment event if the balance impacts of rerating and previous rating are equivalent. To determine if the balance impacts are equivalent, BRM compares the values of certain balance impact fields for

rerating with previous rating, such as the balance element ID and balance group. If the values are different, BRM treats the rerated balance impacts as unique and generates rerating adjustment events.

You can customize how BRM determines whether the balance impacts of rerating and previous rating are equivalent by modifying the event balance impact fields that are used for comparison. For more information, see "Determining Whether Balance Impacts of Rerating and Previous Rating Are Equivalent".

## How BRM Generates Rerated Events

BRM rerates events in chronological order:

- Usage events are rerated in order starting with the earliest usage event.
- For rollover, fold, and cycle discount events, BRM generates new events and applies the balance impacts according to the time that billing was run for the associated cycle.
- For cycle fee events, BRM generates new events and applies the balance impacts according to the time that the accounting cycle ends or when the charge offer is purchased or canceled.

Rollover, fold, cycle discount, and cycle fee events are generated in the order in which they logically occur: for example, billing-time discount events are generated before rollover events.

Rerating generates events in the following ways:

- By rerating the original event

  This process passes the event being rerated to the rating opcode. Rerating generates an adjustment event to apply the balance impacts of rerating. This is the most basic rerating process.

- By reapplying business logic based on information in the original event

  This process passes information from the event being rerated to the opcode that generated the event. The opcode may or may not generate a new event. This process is used when the event attributes may be different after rerating. For example, a charge offer's purchase fee may have been changed since the purchase fee event was generated. To reapply the purchase fee, rerating passes the charge offer information in the original bundle-purchase event to the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL opcode, which generates a new purchase fee event, if applicable.

  > **Note:**
  >
  > In some cases, purchase and cancellation events are not available during rerating. In such cases, BRM rerates the purchase-fee or cancellation-fee event. For more information, see "Events That Are Not Rerated".

- By reapplying business logic independent of the original event

  This process performs business logic that may generate new events even if there is no existing event. With this process, information in the original event is not used. Instead, the business logic that generated the original event is reapplied and a new event is generated, if appropriate.

  For example, when rerating an account for a specific period, if the events selected for rerating are associated with charge offers that have a cycle fee, rerating calls the cycle fee opcode and generates a new cycle fee event based on the charge offer's current rates.

This is done independent of any existing cycle fee event. If there is an existing cycle fee event, it is replaced by the corresponding new cycle fee event generated by rerating.

Table 1-1 shows the rerating process for specific event types.

**Table 1-1 Rerating Process and Event Type**

| Event Type | Order of Rerating | How Rerated |
|---|---|---|
| Usage | Chronological, based on event time | Rerate the original event. |
| Cycle fee | Generated according to the time that the charge offer is purchased or canceled or when the associated accounting cycle starts or ends, and based on the logical order of events | Reapply business logic independent of the original event. |
| Rollover | Generated according to the time that the associated billing cycle ends, and based on the logical order of events | Reapply business logic independent of the original event. |
| Fold | Generated according to the time that the associated billing cycle ends, and based on the logical order of events | Reapply business logic independent of the original event. |
| Billing-time discount | Generated according to the time that the associated billing cycle ends, and based on the logical order of events | Reapply business logic independent of the original event. |
| Purchase | Chronological, based on event time | Reapply business logic based on information in the original event. |
| Cancellation | Chronological, based on event time | Reapply business logic based on information in the original event. |

# About Rerating Unbilled Events

When unbilled events are rerated, a shadow adjustment event is created. This event is called a *shadow event* because its balance impact is added to the original event's bill item rather than to an adjustment item.

## Rerating Unbilled Usage Events

When unbilled usage events are rerated, the shadow adjustment event fully negates the original balance impacts and applies new balance impacts for the rerated amount.

## Rerating Unbilled Cycle Events

When unbilled cycle events are rerated, the shadow adjustment event fully negates the original balance impacts. Then a new cycle event is created that applies the rerated balance impacts. The new cycle event is of the same event type as the original event.

> **Note:**
>
> When the purchase start time is the same as the current accounting cycle end date, the cycle forward fee balance impact is applied to the next month's bill instead of the current month's bill when billing is run after rerating is performed.
>
> For example, suppose an account is created on January 1 with a cycle forward fee of $20 and the purchase start time is deferred by 1 month (set to the accounting cycle end date). On February 1, when **pin_rerate** is run from January 1, because the purchase start time is the same as the accounting cycle end date, rerate internally triggers automatic billing. The billing process changes the status of the bill item to open. Because the bill item status is now open, rerating applies the rerate adjustments to the next bill. When regular billing is run on February 2, the cycle fee is not applied to the January bill; instead, it is applied to the February bill. For more information about rerating billed events, see "About Rerating Billed and Posted Events".

# About Rerating Billed and Posted Events

When you rerate events that are already billed and unbilled events that are already posted to the G/L, the results of rerating are applied as adjustments to the next bill, which is the bill for the current cycle. The balance impacts are applied to the adjustment item.

## Rerating Billed and Posted Usage Events

When usage events that are billed or posted to the G/L are rerated, an adjustment event fully negates the original balance impacts and applies new balance impacts for the rerated amount.

## Rerating Billed and Posted Cycle Events

For billed cycle fee, fold, rollover, and cycle discount events, an adjustment event is created that fully negates the original balance impacts. The adjustment is applied to the current billing cycle and does not impact the original event's items. A new cycle event is then created that applies the rerated balance impacts. The new cycle event is of the same event type as the original event and is applied to the current cycle.

When billing is run, if a noncurrency balance element has a zero balance, no cycle event is generated for that balance element. For example, if there are no remaining free minutes to roll over at billing time, no rollover event is created. However, if rerating creates a nonzero amount for a balance element that was previously zero when billed, the rerating process generates the appropriate cycle event.

# About Rerating and Pricing Changes

Changes to account charge offers are audited. If you change a charge offer attribute after events are rated, and then rerate those events, the rerating process uses only the current account subscription data.

# Events That Are Not Rerated

Usually, purchase-fee events (**/event/billing/product/fee/purchase**) and cancellation-fee events (**/event/billing/product/fee/cancel**) are not directly rerated. Instead, information in the

original bundle purchase event (**/event/billing/deal/purchase**) and product cancellation event (**/event/billing/product/action/cancel**) is resent to the PCM_OP_SUBSCRIPTION_PURCHASE_DEAL and PCM_OP_SUBCRIPTION_CANCEL_PRODUCT opcodes respectively. The opcodes generate new fee events, if applicable.

In some cases, the purchase-fee and cancellation-fee events are directly rerated because the purchase and cancellation events may not be available for rerating: for example, when rerating deferred purchase fees or when using selective rerating in which only purchase-fee or cancellation-fee events are specified.

The following events are not rerated but are reapplied because they were not originally generated by rating:

- **/event/billing/adjustment**
- **/event/billing/charge**
- **/event/billing/cycle/tax**
- **/event/billing/debit**
- **/event/billing/dispute**
- **/event/billing/item**
- **/event/billing/payment**
- **/event/billing/refund**
- **/event/billing/reversal**
- **/event/billing/settlement**
- **/event/billing/writeoff**

## About Rerating Events for Account Sharing Groups

When rerating accounts and bill units in a hierarchy, sponsor, or discount sharing group, BRM does not automatically rerate subordinate and member accounts and bill units. To rerate subordinate and member accounts and bill units, you must specify them when you select the accounts for rerating. However, when BRM rerates an event for a service associated with sponsorship, rerating impacts the appropriate sponsoring or sponsored account's balance.

### About Rerating Events That Impact Sponsored Balance Elements

If the balance impacts of an event are sponsored by another account, the balance impacts of rerating are applied to the sponsoring account. For example, account A pays for Internet access for account B. If account B's events are rerated, the balance impacts of rerating the Internet usage events are applied to account A's balance.

Likewise, if a rerated event impacts a balance element that is shared with another account, the balance element in the sponsored account is adjusted. For example, account A gets 1 frequent flyer mile for every dollar that account B spends. If account B's events are rerated and its current balance is reduced from $100 to $50, account A's frequent flyer miles are adjusted from 100 to 50.

If a rerated, sponsored event impacts a balance element that is shared by more than two accounts, rerating adjusts only the balances in the rerated account and the sponsoring account. For example, account A shares free minutes with accounts B and C. If account B and account C both make calls, and then account B is rerated, any impact on the free minutes

account B used is applied to account A. However, that free-minute balance impact in account A is not carried over to account C, even though account C shares those balance elements.

# BRM Functionality Affected by Rerating

When you rerate events, the following areas in BRM are affected:

- General ledger entries. See "Determining the G/L Entry for an Event".
- Invoices. See "Displaying Shadow-Event Rerating Details on Invoices".

## Determining the G/L Entry for an Event

The G/L entry is determined at the time of rating. If you rerate as a result of pricing changes, the G/L entry could change. Whether you record a shadow event or an adjustment event, you must determine the correct balance to be posted in the G/L.

When recording a shadow event, the G/L ID of the rerating balance impacts have the same G/L ID as the original event's balance impacts. If you changed the G/L ID after the original event was rated, the balance impacts of rerating use the new G/L ID.

When recording an adjustment, you can configure the G/L ID to use. For example, you can use the same G/L ID as the original event, a new G/L ID, or an adjustment G/L ID (a separate G/L ID bucket to record all adjustments as part of rerating). The adjustment G/L ID can be the same or different than the G/L ID used for regular (not rerated) event adjustments.

## Displaying Shadow-Event Rerating Details on Invoices

When recording a shadow event, you can customize your invoice to either show the details of rerating or show the result of rerating only in an end balance.

To customize your invoice:

1. Open the *BRM_home***/sys/cm/pin.conf** file.
2. Set the **show_rerate_details** entry:
   - To show the details of rerating, set the entry to **1**.
   - To show the result of rerating only in an end balance, set the entry to **0.** The default is **0**.

   For example:

   ```
   - fm_inv_pol show_rerate_details 0
   ```
3. Save and close the file.
4. Stop and restart the CM.

# Determining Whether Balance Impacts of Rerating and Previous Rating Are Equivalent

When usage events are rerated, an adjustment event is generated to apply the balance impacts of rerating. The total balance impacts of the adjustment event is the difference between the previous rating and rerating results.

When rerating produces the same result as previous rating, creating an adjustment event is not necessary. However, because an event can include multiple balance impacts, BRM must ensure that the balance impacts in each rating process are actually equivalent.

To determine whether the balance impacts of rerating and previous rating are equivalent, BRM checks the following values of the balance impacts in the event. If any of these values is different for rerating than it is for previous rating, BRM treats the associated balance impact as being unique and generates an adjustment event that includes the balance impact:

- The ID of the balance element being impacted
- The ID of the balance group being impacted
- The G/L ID
- The tax code
- The charge offer used to rate the event

The list of default balance impact fields used to compare the balance impacts of events' rerating and previous rating results are stored in the **/config/rerate_flds/compare_bi** object in the BRM database. You can modify the balance impact fields stored in the **/config/rerate_flds/compare_bi** object to customize how BRM determines whether balance impacts are equivalent.

For example, if you remove the charge offer field and the total rerating adjustment is zero, adjustment events are not generated even when the charge offer used to rerate events is different than the charge offer previously used (providing the other balance impact fields are the same). Or, if you specify additional balance impact fields, adjustment events are generated when the values of those fields differ. For example, you can specify the impact category field to record the rerating adjustment when calls are made from different locations even though rerating results in the same total charge.

To customize the list of event balance impact fields that determine whether rerated and previously rated balance impacts are equivalent, see "Specifying How to Compare Balance Impacts When Creating Adjustment Events".

## Specifying How to Compare Balance Impacts When Creating Adjustment Events

BRM checks the **/config/rerate_flds/compare_bi** object during rerating. If the balance impact fields in this object have the same values in the event for both rerating and previous rating, and the total rerating adjustment is zero, the results of rerating and previous rating are considered equivalent and no adjustment event is created.

> **Note:**
>
> - The **load_pin_rerate_flds** utility overwrites existing balance-impact comparison fields. If you are updating balance-impact comparison fields, you cannot load new fields only: you must load complete sets of the balance-impact comparison fields when you run the **load_pin_rerate_flds** utility.
> - The **load_pin_rerate_flds** utility uses a configuration file (**pin.conf**) located in the same directory to connect to the BRM database. Edit the configuration file to connect to your BRM database.
> - The **load_pin_rerate_flds** utility loads extraction keys that define custom **pin_rerate** parameters. You cannot use the same file to specify extraction keys and balance-impact comparison fields; you must run **load_pin_rerate_flds** with separate files for each configuration.

To customize the fields used to compare the event balance impacts of rerating with previous rating:

1. Open the *BRM_home***/sys/data/config/pin_rerate_compare_bi.xml** file.

   Add a **RerateCompareBalImpacts** element for each event balance impact field. The following balance impact fields are mandatory:

   - PIN_FLD_RESOURCE_ID
   - PIN_FLD_BAL_GRP_OBJ
   - PIN_FLD_GL_ID
   - PIN_FLD_TAX_CODE

   > **Note:**
   >
   > PIN_FLD_PRODUCT_OBJ is specified in the **pin_rerate_compare_bi.xml** file by default, but it is an optional field.

   You can specify any additional field from the **/event** object's PIN_FLD_BAL_IMPACTS array.
2. Save and close the file.
3. Run the following command, which loads the contents of the XML file into the **/config/rerate_flds/compare_bi** object:

   ```
   load_pin_rerate_flds pin_rerate_compare_bi.xml
   ```

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

   ```
   load_pin_rerate_flds BRM_home/sys/data/config/pin_rerate_compare_bi.xml
   ```
4. Stop and restart the CM.
5. To verify that the balance-impact comparison fields were loaded, display the **/config/rerate_flds/compare_bi** object by using Object Browser, or use the **robj** command with the **testnap** utility.

# How BRM Tracks Rerated Events

This section is useful if you write custom code that requires accessing events that were rerated.

To back out original rated events, BRM generates adjustment events (**/event/billing/adjustment/event**). These events contain a compliment of all balance impacts in the original events to fully negate the events. When an event is rerated, the PIN_FLD_RERATE_OBJ field in the original rated event references this backout adjustment event.

When usage events are rerated, the new balance impacts of rerating are included in the same adjustment event that backs out the original balance impacts. This means the original event references the new event that contains the rerated balance impacts.

When cycle fee, cycle discount, fold, and rollover events are rerated, new cycle events are created to apply the balance impacts of rerating. In this case, there is no relationship between the original events and the new (cycle) events that contains the rerated balance impacts.

Because you can use flexible cycles and multiple discounts, there might be more than one rollover, fold, and discount event to rerate for a given billing cycle:

- A fold event is generated per balance for each balance element that has a fold.
- A rollover event is generated per balance group for each charge offer that has a rollover.
- A discount event is generated for each cycle discount.

2

# About Real-Time Rerating Pipelines

Learn how events are processed by Oracle Communications Billing and Revenue Management (BRM) real-time rerating pipelines as well as how to configure real-time rerating pipelines.

Topics in this document:



## About Real-Time Rerating Pipelines

When you run the **pin_rerate** utility to rerate events rated in real-time and events rated in a batch pipeline, you run the **pin_rerate** utility to send pipeline-rated events to a real-time rerating pipeline to be rerated.

The Connection Manager (CM) sends pipeline-rated events in the form of an flist to the NET_EM pipeline module. The NET_EM module transfers the event to a real-time rerating pipeline for rerating.

The real-time rating opcodes add to the flist all the enrichment data needed by the real-time rerating pipeline to rate the event. For example, if the real-time rating opcodes determine that a discount should be applied, it adds the discount information to the flist.

Similar to a batch rerating pipeline, a real-time rerating pipeline performs both rating and discounting functions.

To rerate events, the real-time rerating pipeline gets the new pricing information from the Pipeline Manager database. Certain configuration data, such as currency and noncurrency balance element information, are obtained from the BRM database.

## Overview of Event Processing in the Real-Time Rerating Pipeline

The real-time rerating pipeline processes events in the following way:

1. The INP_Realtime input module converts the flist received from the NET_EM module to event data record (EDR) format and creates an EDR container.

   > **Note:**
   >
   > You can also create and run an iScript to manipulate data in the EDR container before it is sent to the pipeline modules.

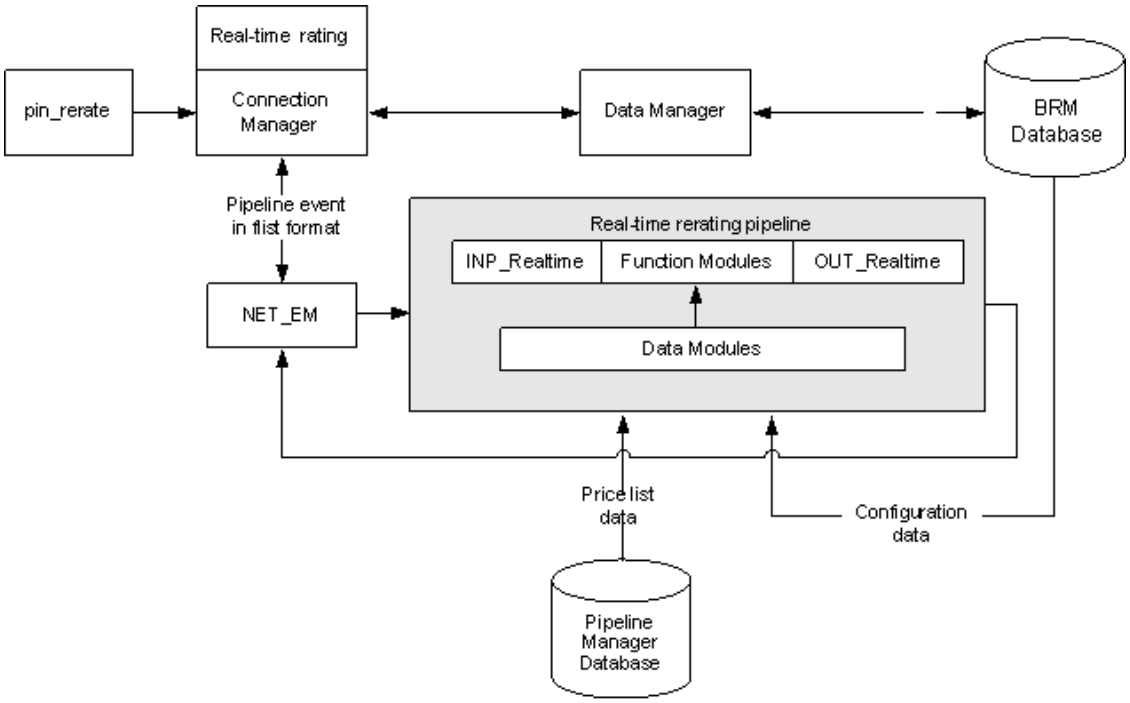
2. The pipeline function modules calculate the new balance impacts by using the new pricing and discounting information and then add the balance impacts to the EDR container.

3. The OUT_Realtime module converts the EDR back into flist format and sends the flist to NET_EM.
4. NET_EM sends the flist with the new balance impacts back to the CM.
5. BRM updates the account's balances and records the event in the BRM database.

For detailed information about configuring the real-time rerating pipeline, see "Configuring Rerating of Pipeline-Rated Events in the Real-Time Rerating Pipeline".

Figure 2-1 shows the data flow for rerating pipeline-rated events by using the real-time rerating pipeline.

**Figure 2-1 Rerating Data Flow for Pipeline-Rated Events**




# About Transaction Management for the Real-Time Rerating Pipeline

The real-time rerating pipeline does not use the Transaction Manager (TAM) module. Instead, transaction handling is provided by the CM. When a rerating transaction fails, the CM rolls back the transaction and restores all the account balances in the BRM database. For this reason, function modules and iScripts used by the real-time rerating pipeline are stateless.

> **Note:**
>
> If you use iScripts for custom processing, ensure that the iScripts are stateless.

# Configuring Rerating of Pipeline-Rated Events in the Real-Time Rerating Pipeline

To configure rerating of pipeline-rated events in the real-time rerating pipeline, perform the following tasks:

- Configure the real-time rerating pipeline. See "Configuring a Real-Time Rerating Pipeline".

> **Note:**
>
> When you configure pipelines in the **Pipelines** section, you must add the pipelines to the IFW_PIPELINE table. Otherwise, you will receive an error at system startup. For more information, see "Configuring the Real-Time Rerating Pipelines in the IFW_PIPELINE Table".

- Configure the CM to send rerate requests to the NET_EM module.
- Configure NET_EM to route rerate requests to the real-time rerating pipeline.

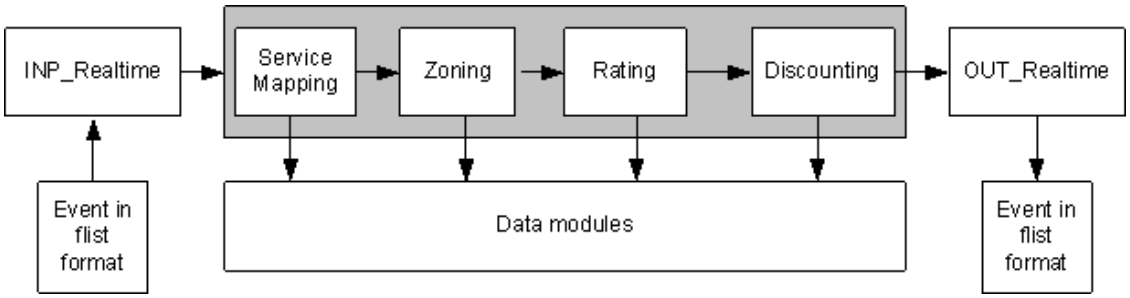
## Configuring a Real-Time Rerating Pipeline

A real-time rerating pipeline is configured similarly to a batch rerating pipeline, but with fewer preprocessing and post-rating modules. This is because most of the enrichment data is provided in the input flist received from the CM.

You typically create a separate instance of Pipeline Manager for real-time rerating. The default registry file is *Pipeline_home***/conf/wirelessRealtime.reg**.

Figure 2-2 shows the real-time rerating pipeline architecture.

**Figure 2-2 Real-Time Rerating Pipeline Architecture**




## Configuring Multiple Real-Time Rerating Pipelines

To improve performance or scalability, you configure multiple instances of real-time rerating pipelines. For example, you can increase performance by configuring multiple pipelines to process rerate requests in parallel or by configuring multiple real-time rerating pipelines to process different rerate requests: for example, by configuring one pipeline that rerates only GPRS events and another that rerates only GSM events.

## Configuring the Real-Time Rerating Data Pool

Configure the following data modules:

- DAT_Zone
- Rateplan
- DAT_Calendar
- DAT_TimeModel
- DAT_PriceModel
- Dayrate
- DAT_Currency
- DAT_USC_Map

## Configuring the Modules in the Real-Time Rerating Pipeline

Configure the following modules in the real-time rerating pipelines:

- Configure the INP_Realtime input module. Specify the flist-to-EDR mapping file used for rerating in the **OpcodeMapping** registry entry. For example:

  **OpcodeMapping = ./formatDesc/Formats/Realtime/rate_event.xml**

- Configure the NET_EM module:
  - Configure NET_EM to manage data between the CM and Pipeline Manager and configure the CM to send rerate request to the NET_EM module.
  - Configure NET_EM to route rerate requests. See "Configuring NET_EM to Route Rerate Requests Based on the Event Field Value".
- Configure the output module to add any customizations to the output flist.
- Configure these function modules:
  - FCT_ServiceCodeMap
  - FCT_CustomerRating
  - FCT_PreRating
  - FCT_IRules
  - FCT_USC_Map
  - FCT_RSC_Map
  - FCT_MainRating
  - FCT_Dayrate
  - FCT_RateAdjust
  - FCT_Rounding
  - FCT_DiscountAnalysis
  - FCT_Discount

## Configuring the Real-Time Rerating Pipeline to Set Charge Offer Validity Periods

If your charge offers are configured to start when they are first used, configure the ISC_FirstProductRealtime iScript in the real-time rerating pipeline.

When charge offers start on first usage, the real-time rerating pipeline adds the account's first-usage charge offer and discount information to the EDR. ISC_FirstProductRealtime sets the validity period in the BRM database for those charge offers that were used to rate the event and that start on first usage. This triggers any purchase and cycle fees.

## Configuring the Real-Time Rerating Pipelines in the IFW_PIPELINE Table

Pipeline Manager stores information about pipelines in the IFW_PIPELINE table. The pipelines that are preconfigured in the **Pipelines** section of the default registry file (*Pipeline_home***/conf/wirelessRealtime.reg**) are inserted into the IFW_PIPELINE table during Pipeline Manager installation.

> **Note:**
>
> - If you are *not* using the default registry file, and you have configured new real-time rerating pipelines, you must manually insert the pipelines into the IFW_PIPELINE table by using SQL commands. Otherwise, you will receive an error at system startup.
> - If you *are* using the default registry file and have changed the default pipeline names or you have configured additional pipelines, you must manually insert the new pipelines into the IFW_PIPELINE table.

The following example shows SQL commands to insert RealtimeReratingGSM and RealtimeReratingGPRS pipelines into the IFW_PIPELINE table:

```
% sqlplus pin/password@databaseAlias

SQL>INSERT INTO IFW_PIPELINE ( PIPELINE, NAME, EDRC_DESC ) VALUES
( 'RealtimeReratingGSM', 'GSM Realtime Rerating Pipeline', 'ALL_RATE');

SQL>INSERT INTO IFW_PIPELINE ( PIPELINE, NAME, EDRC_DESC ) VALUES
( 'RealtimeReratingGPRS', 'GPRS Realtime Rerating Pipeline', 'ALL_RATE');

SQL>commit;
```

3

# About Rerating Pipeline-Rated Events

You can rerate events that were originally rated in batch by Oracle Communications Billing and Revenue Management (BRM) Pipeline Manager.

Topics in this document:



## Overview of Rerating Events

It is possible to discover pricing or rating configuration errors after events have been rated. For example, a sample review of invoices might reveal a price configuration error. In addition to correcting the price configuration, you might need to rerate events that were rated with the incorrect configuration.

Use the following components to rerate events that were originally rated by Pipeline Manager:

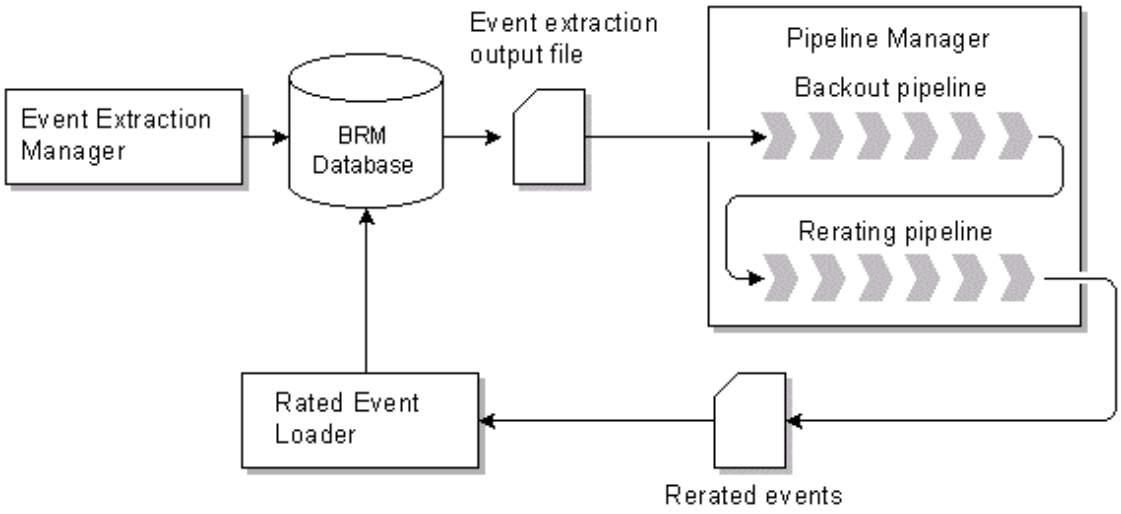
1. Use the *Event Extraction utility* to obtain data from the BRM database about the events you must rerate. This utility writes event data to a file.
2. Use *Pipeline Manager* to process the file that was created by the Event Extraction utility. There are two steps:
   a. Restore discount and aggregation data to the state it was in before rating the events. For example, if you use volume discounting based on calls made, you can adjust the number of calls made.
   b. Rerate the events using the new rating data. This process creates an output file.
3. Use the *Rated Event Loader (RE Loader)* to load the output file generated by Pipeline Manager into the BRM database.

Pipeline Manager only rerates events that it has previously processed. When you use pipeline rerating, there might be events that are not rerated. For example, if you retrieve events for rerating based on the account, any events belonging to the account that were generated in BRM and not previously processed by Pipeline Manager are not rerated. Ensure that all events are rerated by also running the BRM rerating utility (**pin_rerate**).

> **Note:**
>
> When you rerate wireless events by using Pipeline Manager, you rerate only EDRs that have been rated by Pipeline Manager. Events that have been rated in real time are rerated by using the **pin_rerate** utility.

Figure 3-1 shows the rerating process.

**Figure 3-1 Rerating Process**




# About Extracting Events from the BRM Database

The Event Extraction Manager searches for and extracts events meeting a specified search criteria and writes the results to an output file. It does not modify or process any of the events, nor does it lock accounts in your database. This enables BRM and other applications to safely access accounts in the database while you are extracting events.

To extract events, you perform the following steps:

1. Edit the event search file to specify which events to extract. You can specify events based on many different event attributes, for example, date, account, charge offer, charge, and service.
2. Configure and run the **pin_event_extract** utility to create the event extract output file. This file contains the list of events to rerate in EDR (event data record) format, and is used as the input for the Pipeline Manager rerating process.

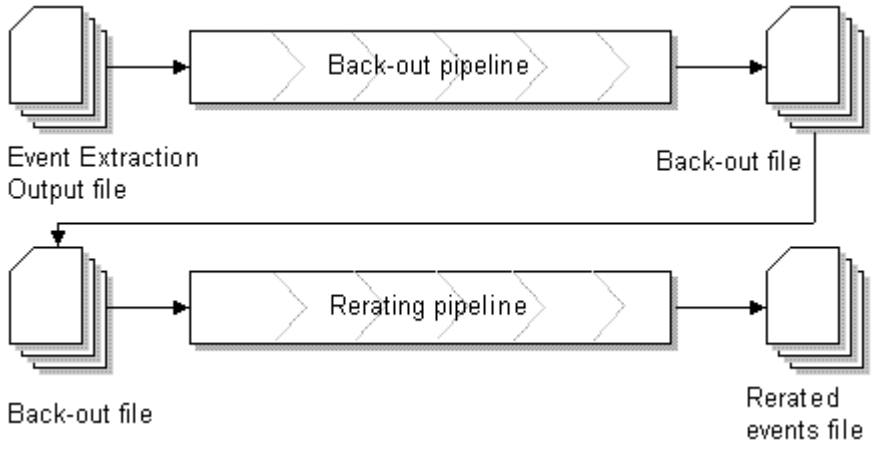
# About Rerating Events with Pipeline Manager

To rerate events with Pipeline Manager, you perform the following steps:

1. Correct the rating configuration that caused the non-valid rating.
2. Configure the back-out pipeline. This pipeline uses the event extract output file as input. It backs out the existing balance impacts from discount accounts, aggregates the results, and outputs a back-out file.
3. Configure the rerating pipeline. This pipeline uses the back-out file as input. It rerates the events using the corrected rating configuration. The rerating process is the same as the rating process, however, the output file also includes the delta of the old and new balance impacts. Rated Event Loader loads the file into the BRM database.

Figure 3-2 shows the rerating steps described.

**Figure 3-2 Pipeline Manager Rerating Process**




## About Loading Rerated Events into the BRM Database

You load rerated events into the BRM database by running the Rated Event Loader (RE Loader). RE Loader identifies events that have already been billed, ensuring that account balances are accurately updated. See "Adjusting Account Balances".

To load rerated events into the BRM database, you perform the following steps:

1. Set up the RE Loader processing directories.
2. Configure RE Loader to load the rerated event types by configuring the *REL oader_home*/**Infranet.properties** file.
3. Run RE Loader.

### Adjusting Account Balances

RE Loader determines if each event has been billed. For events that *have* been billed, RE Loader creates an adjustment event (**/item/adjustment**) with the appropriate balance impact and sets the event type to **/event/billing/adjustment/event**. It then loads the events and updates the account balances. RE Loader updates the event as having been rerated and keeps a history of each update.

# About Back-Out-Only Rerating

You use back-out-only rerating to reprocess events that were originally rated by Pipeline Manager. It enables you to back out only the event balance impacts without rerating the events.

For example, you might want to use back-out-only rerating when you find that call detail records (CDRs) rated and recorded in the BRM database contain incorrect values. In such cases, you can use back-out-only rerating to back out the event balance impacts without rerating the event.

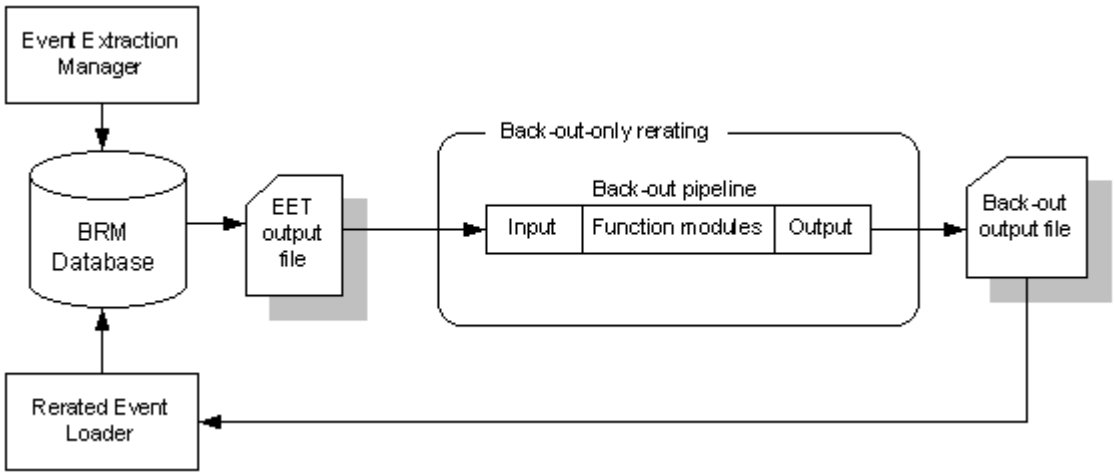
The output file created by backout-only rerating is loaded into the BRM database by the Rated Event (RE) Loader.

To process events for back-out-only rerating:

1. Use the Event Extraction Manager to extract the events from the BRM database. See "About Extracting Events for Back-Out-Only Rerating".
2. Use the backout pipeline to process the output file generated by the Event Extraction Manager. See "About Configuring the Backout Pipeline for Back-Out-Only Rerating".
3. Use the Rated Event Loader to load the output file generated by the backout pipeline into the BRM database. See "About Loading Rerated Events into the BRM Database".

Figure 3-3 shows the back-out-only rerating process.

**Figure 3-3 Back-Out-Only Rerating Process**




## About Extracting Events for Back-Out-Only Rerating

Use the Event Extraction Manager to find and extract events from the BRM database for backout-only processing. The Event Extract Tool (**pin_event_extract**) generates an output file containing the events in EDR format.

To extract events for back-out-only rerating, run **pin_event_extract** with the **-e** parameter.

> **Note:**
>
> If you do not use the **-e** parameter, Pipeline Manager sends the EDRs in the output file to the rerating pipeline for rerating.

## About Configuring the Backout Pipeline for Back-Out-Only Rerating

You use the backout pipeline to process the EDRs extracted by the Event Extraction Manager. The backout pipeline backs out the event balance impacts and writes the events to an output file.

The backout pipeline generates output files for both normal rerating and back-out-only rerating:

- In normal rerating, the output file is processed by the batch-rerating pipeline.
- In back-out-only rerating, the output file is processed by RE Loader.

Both types of rerating require different file formats and output grammar.

RE Loader loads events of different types from separate directories. For this reason, with back-out-only rerating, the backout pipeline sends the EDRs to different output streams based on the event type. For example, the backout pipeline writes all GSM events in one file, and all GPRS events in another file, and sends them to separate output directories.

To configure back-out-only rerating:

1. Configure the OUT_GenericStream module to generate the output file using the output grammar for back-out-only rerating and to send the output files to different directories based on the event type. See "Configuring the OUT_GenericStream Module".
2. Configure RE Loader to load the back-out output file into the BRM database. See "About Loading Rerated Events into the BRM Database".

## Configuring the OUT_GenericStream Module

The OUT_GenericStream module formats the output for back-out-only rerating based on the grammar file specified in its registry. The EXT_OutFileManager sends the data to the output directories based on the event type specified in its registry. You configure the EXT_OutFileManager module inside the OUT_GenericStream module.

To configure the OUT_GenericStream and EXT_OutFileManager modules:

- Set the **Grammar** entry to:

  **`./formatDesc/Formats/Solution42/V670_EVENT_LOADER_OutGrammar.dsc.`**

- Set the **OutputPath** entry to specify the output directory. For example, for GSM events, set **OutputPath** to:

  **`./data/backout/out/gsm/telephony.`**

Sample OUT_GenericStream registry for GSM events:

```
BackOutOnlyTELOutput
{
     ModuleName = OUT_GenericStream
     ProcessType = ALL_BCKOUT
     EventType = /event/delayed/session/telco/gsm
     Module
     {
          Grammar = ./formatDesc/Formats/Solution42/V670_EVENT_LOADER_OutGrammar.dsc
          DeleteEmptyStream = True
          OutputStream
          {
               ModuleName = EXT_OutFileManager
               Module
               {
                    OutputPath = ./data/backout/out/gsm/telephony
                    OutputPrefix = test_TEL
                    OutputSuffix = .out
                    TempPrefix   = .
                    TempDataPath    = ./data/backout/out/gsm/telephony
                    TempDataPrefix  = tel.tmp.
                    TempDataSuffix  = .data
                    Replace = TRUE
               }
          }
     }
}
```

Sample OUT_GenericStream registry for GPRS event:

```
BackOutOnlyGPRSOutput
{
     ModuleName = OUT_GenericStream
     ProcessType = ALL_BCKOUT
     EventType = /event/delayed/session/gprs
     Module
     {
          Grammar = ./formatDesc/Formats/Solution42/V670_EVENT_LOADER_OutGrammar.dsc
          DeleteEmptyStream = True #defaults to true
          OutputStream
          {
               ModuleName = EXT_OutFileManager
               Module
               {
                    OutputPath = ./samples/wireless/data/backout/out/gprs
                    OutputPrefix = test_GPRS
                    OutputSuffix = .out
                    TempPrefix   = .
                    TempDataPath    = ./samples/wireless/data/backout/out/gprs
                    TempDataPrefix  = gprs.tmp.
                    TempDataSuffix  = .data
                    Replace = TRUE
               }
          }
     }
} # end of BackOutOnlyGPRSOutput
```

> **Note:**
>
> To ensure output file integrity, specify a unique combination of OutputPath, OutputSuffix, and OutputPrefix values for each output stream defined in the registry.

# About Synchronizing Rating and Loading Applications

When you use Pipeline Manager to rerate wireless events and load those events using RE Loader, you must synchronize the following applications:

- Pipeline Manager
- Event Extraction
- Rated Event Loader

You must synchronize these applications for two reasons:

- RE Loader locks the BRM database tables that it loads into, making them inaccessible to other applications during the loading process. Event Extraction Manager returns an error if it cannot access the database tables.
- If any events have not been rated and loaded before Pipeline Manager starts rerating, the account balances can be incorrectly updated when those events are processed after rerating is complete.

You must synchronize these applications manually by stopping and restarting them in a specific order:

1. Ensure that all events have been rated by Pipeline Manager.
2. Ensure that all rated events have been loaded by RE Loader.

There should be no remaining unrated files in the pipeline output directories. If errors occurred during loading that require you to reload events, perform those tasks before continuing.

3. Stop the rerating pipeline.
4. Stop the Batch Controller, thereby disabling any scheduled RE Loader processes.
5. Run the Event Extraction utility.
6. Start the backout pipeline to remove all the erroneous impact balances from the discount accounts.
7. Stop the backout pipeline and restart the rerating pipeline to rerate the events. The rerating pipeline can be restarted using the Pipeline Manager EventHandler.
8. If you configured RE Loader to run *manually*, run the RE Loader **pin_rel** utility. If errors occur during loading, correct them and reload the events before continuing.
9. If you configured RE Loader to run *automatically*, enable the REL handler that runs **pin_rel** in the Batch Controller configuration file by doing the following:
   - Stop and restart the Batch Controller to run **pin_rel** and load the rerated events. If errors occur during loading, correct them and reload the events before continuing.
   - Disable the REL handler that runs **pin_rerel** in the Batch Controller configuration file.
10. Enable RE Loader in the Batch Controller configuration file.

4

# Using Event Extraction Manager

You use the Oracle Communications Billing and Revenue Management (BRM) Event Extraction Manager to extract prerated events for rerating.

> **Note:**
>
> Event Extraction Manager is packaged with Rated Event (RE) Loader.

Topics in this document:



Before extracting events, you must know the following:

- Basic BRM concepts
- BRM system architecture
- How to create and edit BRM configuration files

## About Event Extraction Manager

Event Extraction Manager is the first application used in the rerating process. You use it to find and extract events from the BRM database that must be rerated by Pipeline Manager and then reloaded into the BRM database by Rated Event (RE) Loader.

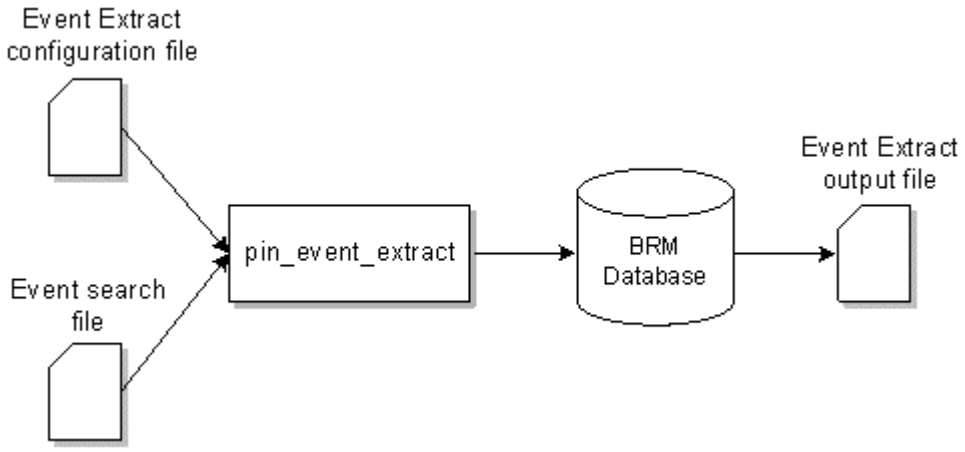
Event Extraction Manager does not modify or process events, nor does it lock accounts in your database. This enables BRM and other applications to safely access accounts in the database while you are extracting events.

Event Extraction Manager consists of the following components:

- The event extract configuration file (**pin.conf**), which specifies how the **pin_event_extract** utility connects to your BRM system and the size and name of the event extract output file
- The **pin_event_extract.cfg** file, which specifies the event search criteria
- The **pin_event_extract** utility, which searches the BRM database for events meeting the search criteria and writes the results to an output file
- The event extract output file, which contains the list of events to rerate

Figure 4-1 shows Event Extraction Manager.

**Figure 4-1 Event Extraction Manager**




## About the Event Extract Configuration File

The event extract configuration file (**pin.conf**), which specifies how the **pin_event_extract** utility connects to your BRM system, and the size and name of your event extract output file.

If you are extracting events from a multischema system, you must modify this file for each database schema in your system. For information, see "Extracting Events from a Multischema System".

For information on how to create this file, see "Configuring Connection and Output File Parameters".

## About the pin_event_extract.cfg File

The event search file (*BRM_home*/**apps/pin_event_extract/pin_event_extract.cfg**) specifies the criteria that the **pin_event_extract** utility uses to search for events in your BRM database. You can search for events based on a wide variety of criteria, including account number, charge offer name, and impact category.

For information about the event search criteria, see "Event Search Criteria".

For information on creating a **pin_event_extract.cfg** file, see "Specifying Which Events to Extract for Rerating".

## About the pin_event_extract Utility

The **pin_event_extract** utility builds a search query with the criteria specified in the **pin_event_extract.cfg** file and then run it against the specified database schema. The search query selects only events that have a balance impact type of pipeline-prerated, indicating that Pipeline Manager has already rated the events.

## About the Event Extract Output File

The **pin_event_extract** utility prints to a file the list of events meeting the search criteria. Results are written in tab-delimited columns conforming to the Oracle CDR format. You send this file directly to Pipeline Manager for rerating.

The default implementation defines the event-field-to-output file mapping for GSM and GPRS events. However, you can create additional categories by modifying the PCM_OP_BILL_POL_CONFIG_EET policy opcode.

When the number of events meeting the search criteria exceeds the specified maximum file size or number of EDRs, the utility generates multiple output files and appends a number, such as **_1**, **_2**, and so on, to each file name.

When you use the sequence number search criteria, the utility automatically appends a sequence number to the file name and ignores the specified maximum file size and number of EDRs.

# Event Search Criteria

Event Extraction Manager searches for events by using the criteria in Table 4-1.

To specify which events to extract:

- To extract events generated for a specific brand, run the **pin_event_extract** utility with the **-b** *BrandName* option.
- To extract events with a specified starting timestamp, run the **pin_event_extract** utility with both the **-f** *ConfigFileName* and **-s** options.
- To extract events using all other criteria, edit the **pin_event_extract.cfg** file.

> **Note:**
>
> The event start time, event end time, and event type criteria are required.

**Table 4-1 Event Search Criteria**

| Criteria | Entry Name | Description | Required |
|---|---|---|---|
| **account number** | `account` | Extracts events generated by the specified account.<br>**Caution:** If you use this criteria, ensure the following:<br>– All events for the specified accounts are available for rerating<br>– All specified accounts reside in the same database schema<br>– The event start time, event end time, and event type criteria are required. | no |

**Table 4-1 (Cont.) Event Search Criteria**

| Criteria | Entry Name | Description | Required |
|---|---|---|---|
| **deal name** | `deal` | Extracts events generated by all charge offers and discount offers associated with the specified bundle.<br>**Caution:** Do not use this criteria with the charge offer or charge search criteria. Doing so prevents the utility from extracting all events associated with the specified bundle. | no |
| **event occurrence start and end times** | `event_start_time_stamp`<br>`event_end_time_stamp` | Extracts events that occurred after the specified starting time stamp and before the specified ending time stamp. The start and end times are inclusive. | yes |
| **event creation start and end times** | `event_created_start_time_stamp`<br>`event_created_end_time_stamp` | Extracts events loaded into the database after the specified creation start time stamp and before the specified creation end time stamp. The start and end times are inclusive. | no |
| **event type** | `event_category` | Extracts the specified event type. By default, this criteria is set to extract the **/event/delayed/session/telco/gsm** event type. | yes |
| **G/L ID number** | `gl_id` | Extracts events with the specified general ledger G/L ID. | no |
| **impact category name** | `impact_category` | Extracts events rated with the specified impact category. | no |
| **item object number** | `item_obj` | Extracts events generated for the specified item. | no |
| **product name** | `product` | Extracts events generated for the specified charge offers.<br>**Caution:** Do not use this criteria with the bundle or charge search criteria. | no |
| **charge name** | `rate_plan` | Extracts events rated by the specified charge.<br>**Caution:** Do not use this criteria with the bundle or charge offer search criteria. | no |

**Table 4-1 (Cont.) Event Search Criteria**

| Criteria | Entry Name | Description | Required |
|---|---|---|---|
| **resource ID** | `resouce_id` | Extracts events with the specified balance element ID. This criteria works best when the balance element ID represents a noncurrency asset such as Internet hours. | no |
| **search event field** | `search_event_type`<br>`search_field`<br>`search_parameter` | Extracts events of type *search_event_type* where *search_field* contains *search_parameter*. For example, you can use this search criteria to extract events based on custom fields.<br>– *search_event_type* specifies the type of event to search. Only events of this type or a subclass of this event type are extracted<br>**Important:** All events of type *search_event_type* must contain the *search_field* . You must ensure that the storable class specified by the event type contains *search_field*.<br>– *search_field* specifies the field in the event used for event extraction. Only those events where *search_field* is equal to *search_parameter* are extracted.<br>– *search_parameter* specifies the value of *search_field*.<br>**Note:** *search_parameter* is used in addition to other search criteria specified in the **pin_event_extract.cfg** file. | no |

**Table 4-1 (Cont.) Event Search Criteria**

| Criteria | Entry Name | Description | Required |
|---|---|---|---|
| **sequence number** | `file_sequence_number` | Extracts events processed with the specified file sequence number. This number is assigned by Pipeline Manager and then added to the event object when RE Loader loads the event into the database. This enables you to extract and rerate events that were processed at a particular time.<br>You can find an event's file sequence number in the **/batch/rel** object.<br>When you use this criteria, the **pin_event_extract** utility automatically appends the file sequence number to the output file name.<br>**Note:** You can specify any sequence number other than 0. RE Loader uses 0 to indicate that Pipeline Manager did not assign a sequence number to the event. | no |
| **service name** | `service` | Extracts events generated for the specified service.<br>**Important:** The specified service must correspond to a unique service type. | no |

# Synchronizing Extraction and Rating Applications

It is important to rate all events before running Event Extraction Manager. If there are events not yet rated when you run Event Extraction Manager, your account balances will not be correct.

Event Extraction Manager and RE Loader use the same status table, REL_EVENT_EXTRACT_SYNC_T, to check if one of the other applications is running. If you attempt to start Event Extraction Manager while RE Loader is running, Event Extraction Manager fails to start and returns an error message.

However, if Event Extraction Manager or RE Loader terminate abnormally, the status table may be incorrectly left in a locked state. You must use the **pin_event_extract** utility's override option before you can extract events.

# Extracting Events

Extracting events includes the following tasks:

# Configuring Connection and Output File Parameters

To configure your connection and output file parameters:

1. Open the event extract configuration file (*BRM_home*/**apps/pin_event_extract/pin.conf**) in a text editor.
2. Edit the connection and log file entries.
3. Specify the EDR output file name:

   ```
   - pin_event_extract filename SOL42_SenderReceiver
   ```

   where *Sender* is the code for the sender of the file, and *Receiver* is the code for the recipient of the file. For example, if the sender's code is D00D1 and the receiver's code is SOL42, replace *SenderReceiver* with **D00D1SOL42**.

   > **Note:**
   >
   > - When you use the sequence number search criteria, the utility automatically appends the file sequence number to the file name.
   > - When the output exceeds the specified file maximums, the utility automatically generates multiple output files and appends a number to each file name.

4. Specify the UTC Time Offset: where UTC Time Offset is the difference between the local time and UTC time:

   ```
   - pin_event_extract UTCoffset Value
   ```

5. Specify the specification version number:

   ```
   - pin_event_extract specvernum VersionNumber
   ```

6. Specify the specification release version number:

   ```
   - pin_event_extract specrelver ReleaseNumber
   ```

7. Specify the maximum number of EDR creation threads:

   ```
   - pin_event_extract num_threads MaximumNumber
   ```

8. Specify the number of EDR records Event Extraction Manager retrieves during each step search:

   ```
   - pin_event_extract step_size SearchNumber
   ```

9. Specify the maximum size, in bytes, of the event extract output file:

   ```
   - pin_event_extract   maxfilesize   FileSize
   ```

When the file size exceeds the specified maximum, the **pin_event_extract** utility generates multiple output files and appends a number, such as **_1**, **_2**, and so on, to each file name.

10. Specify the maximum number of EDRs allowed in each output file.

```
- pin_event_extract    maxEDRs    MaximumEDRs
```

When the number of EDRs exceeds the specified maximum, the **pin_event_extract** utility generates multiple output files and appends a number, such as **_1**, **_2**, and so on, to each file name.

11. Save and close the file.

## Specifying Which Events to Extract for Rerating

The **pin_event_extract** utility finds the events for rerating by creating a search query. You specify the parameters for the search in the *BRM_home***/apps/pin_event_extract/pin_event_extract.cfg** file.

To extract events:

1. Open the *BRM_home***/apps/pin_event_extract/pin_event_extract.cfg** file in a text editor.
2. Set the **event_start_time_stamp** and **event_end_time_stamp** entries.

```
event_start_time_stamp MM/DD/YYYY [hh:mm:ss]
event_end_time_stamp MM/DD/YYYY [hh:mm:ss]
```

where *hh:mm:ss* is the time in 24-hour mode. For example, enter **16:00:00** to specify 4:00 p.m. If you do not specify *hh*:*mm*:*ss*, the time defaults to midnight (00:00:00).

The **pin_event_extract** utility extracts events that occurred between the specified start date and end date (inclusive).

3. Set the **event_category** entry:

```
event_category      category
category            EventTypeName
```

where:

- *category* specifies the type of event to extract. The default implementation includes the **gsm** and **gprs** event categories only, but you can create additional categories by modifying the PCM_OP_SUBSCRIPTION_POL_CONFIG_EET policy opcode.
- *EventTypeName* specifies the name of the event you want extracted. For example, **/event/delayed/activity/gprs**.

> **Note:**
>
> You can list multiple event types on one line by using colons (**:**) or semicolons (**;**) to delimit events.

4. Set your search criteria. The **pin_event_extract.cfg** file includes examples and instructions.

For an overview, see "Event Search Criteria".

5. Save and close the file.

## Extracting Events from Specific Partitions

Events are loaded into database tables that are partitioned by a period such as days, weeks, or months. You can improve event extraction performance by limiting the number of partitions from which events are extracted. To do this, specify the start and end dates of when the events were loaded into the database.

For example, if your database tables are partitioned by months, you can extract events from a single partition by specifying the start and end day of a single month.

To extract events from specific partitions:

1. Open the *BRM_home*/**apps/pin_event_extract/pin_event_extract.cfg** file in a text editor.
2. Set the **event_created_start_time_stamp** and **event_created_end_time_stamp** entries. The **pin_event_extract** utility extracts events that were loaded into the database on or after the start time and on or before the end time.

   ```
   event_created_start_time_stamp MM/DD/YYYY hh:mm:ss
   event_created_end_time_stamp MM/DD/YYYY hh:mm:ss
   ```

   where:

   - *MM/DD/YYYY* is the month, day, and year.
   - *hh:mm:ss* is the time in 24-hour mode. For example, enter *16:00:00* to specify 4:00 p.m.

   > **Note:**
   >
   > Entering only a day returns an error.

## Sample pin_event_extract.cfg File for a Specific Charge

The following sample **pin_event_extract.cfg** file extracts events that meet the following criteria:

- Created between 6:00 p.m. on June 1, 2025, and 6:00 p.m. on August 30, 2025.
- Is the following prerated GSM event: **/event/delayed/session/telco/gsm**.
- Rated with the Free Saturdays charge.

```
event_start_time_stamp 06/01/2025 18:00:00
event_end_time_stamp 08/30/2025 18:00:00

event_category   gsm
gsm              /event/delayed/session/telco/gsm

rate_plan Free Saturdays
```

## Sample pin_event_extract.cfg File for a Specific Sequence Number

The following sample **pin_event_extract.cfg** file extracts events that meet the following criteria:

- Created between 10:00 a.m. on January 1, 2025 and 9:59:59 a.m. on January 31, 2025.

- Represents a prerated GPRS event
- Processed with the 777888999 sequence number

```
event_start_time_stamp 01/01/2025 10:00:00
event_end_time_stamp 01/31/2025 09:59:59
event_type /event/delayed/session/gprs
file_sequence_number 777888999
```

# Running the pin_event_extract Utility

The **pin_event_extract** utility generates a search query with your specified search criteria. See "pin_event_extract" for more information.

To extract events:

1. Ensure that all events that you want to extract have been processed.
2. Ensure that RE Loader is not running.

   For information, see "Synchronizing Extraction and Rating Applications".

   > **Note:**
   >
   > If RE Loader and **pin_event_extract** are run simultaneously, **pin_event_extract** extracts only those records that are completely processed by RE Loader.

3. Run the **pin_event_extract** utility:

   ```
   pin_event_extract -f ConfigFileName
   ```

   where *ConfigFileName* is the name and location of the **pin_event_extract.cfg** file.

# Troubleshooting Event Extraction Errors

Event extraction may fail for several reasons:

- RE Loader is currently running, preventing Event Extraction Manager from starting. Wait for RE Loader to finish loading events before restarting Event Extraction Manager.
- The status table was incorrectly locked, preventing Event Extraction Manager from starting. This can occur when one of the applications terminates abnormally and cannot reset the status table before exiting. To reset the status table:

  > **Note:**
  >
  > Do not use this option unless you are certain that RE Loader is stopped.

  ```
  pin_event_extract -o TRUE
  ```

You can check the event extract log file, created by default in the *BRM_home***/var/pin_event_extract** directory, for information on why extraction failed.

After you fix the problem, you can safely rerun the **pin_event_extract** utility with the same **pin_event_extract.cfg** file.

## Sending the Output File to Pipeline Manager

To begin rerating events, send the event extract output file to Pipeline Manager for processing.

# Extracting Events from a Multischema System

To extract events from a multischema system:

1. On the system on which Event Extraction Manager is installed, open the *BRM_home*/**apps/pin_event_extract/pin.conf**in a text editor.
2. Specify which events to extract by using the **pin_event_extract.cfg** file.
3. Run the **pin_event_extract** utility and fix any errors.
4. Change the connection and output file parameters for the next schema.
5. Run the **pin_event_extract** utility and fix any errors.
6. Repeat steps 4 and 5 until you have extracted events from all schemas.

# Customizing How to Extract Events for Rerating

You can use the PCM_OP_SUBSCRIPTION_POL_CONFIG_EET policy opcode to customize which event fields are extracted for rerating; for example, you can modify which GPRS fields are extracted, or you can define the mapping for a custom event category.

The PCM_OP_SUBSCRIPTION_POL_CONFIG_EET policy opcode defines, for each event category, the event fields that Event Extraction Manager passes to the event extract output file. The default implementation defines the event-field-to-output-file mapping for GSM and GPRS event categories only.

This policy opcode is called by Event Extraction Manager worker threads when the tool returns a list of POIDs that meet the search criteria.

By default, the PCM_OP_SUBSCRIPTION_POL_CONFIG_EET policy opcode takes an event's POID and category name as input and performs the following functions:

- Reads the entire event from the BRM database
- Maps predefined fields for GSM and GPRS events to a record buffer in the order received

> **Note:**
>
> The **pin_event_extract** utility writes the contents of the record buffer to the event extract output file.

You can customize the PCM_OP_SUBSCRIPTION_POL_CONFIG_EET policy opcode to pass extended fields to the event extract output file. For example, you can modify which GPRS fields are passed to the output file, or you can define the mapping for a custom event category.

You can do the following:

- Read any additional event data based on the given POID
- Handle any new input formats: the policy opcode is written for the Oracle CDR forma
- Define the event-field-to-output-file mapping for any new categories

- Write data to the correct position in the record buffer

5

# Configuring Rerating in Pipeline Manager

Learn how to rerate events using the Oracle Communications Billing and Revenue Management (BRM) Pipeline Manager.

Topics in this document:



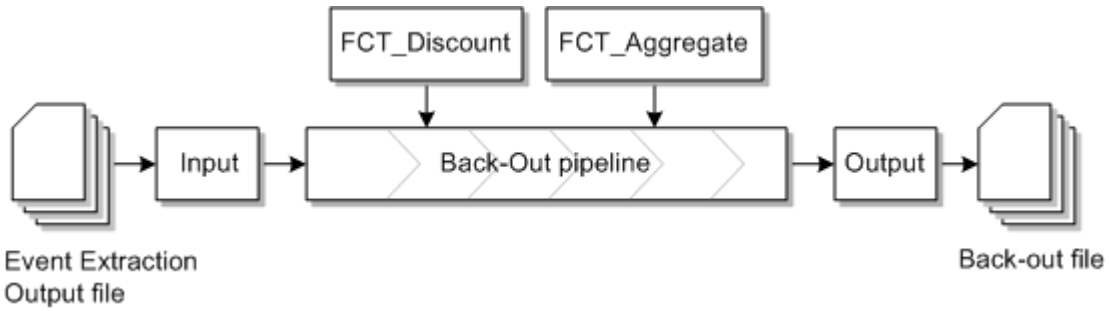
## About the Back-Out Pipeline

The back-out pipeline is a separate pipeline and contains the following modules:

- The FCT_Discount module removes the EDR discount balance impacts from the discount accounts.
- The FCT_AggreGate module calculates the aggregation results and writes them to the back-out result file. It writes one result file for each transaction.

Figure 5-1 shows the back-out pipeline.

**Figure 5-1 Back-Out Pipeline**




## Configuring the Back-Out Pipeline

When you configure the back-out pipeline, you configure the following parameters in the pipeline registry:

- The ISC_AddCBD iScript. You configure this by configuring an instance of the FCT_IScript module.
- The FCT_Discount module. You set the **BackOut** registry entry to TRUE.
- The FCT_AggreGate module. You set the **BackOut** registry entry to TRUE.

Pipeline Manager creates one control file and one result file for each transaction and each aggregation scenario.

## About Starting the Back-Out Pipeline

Before you start the back-out pipeline stop all rating pipelines and ensure that there are no outstanding events being loaded.

You start the back-out pipeline by sending a semaphore:

```
ifw.Pipelines.ALL_BCKOUT.Active = TRUE
```

## About Stopping the Back-Out Pipeline

You stop the back-out pipeline by sending a semaphore:

```
ifw.Pipelines.ALL_BCKOUT.Active = FALSE
```

# About the Rerating Pipeline

The rerating pipeline contains the same function modules as the rating pipeline, except that it does not contain the FCT_CallAssembling and FCT_DuplicateCheck modules. The rerating pipeline locks the appropriate customer accounts until rerating is completed.

> **Note:**
>
> Before you start the rerating pipeline, ensure that you correct all configuration data in the Pipeline Manager database, such as the charge that caused the incorrect calculation.

After the EDRs are re-rated with the updated configuration data, the FCT_BillingRecord module writes to the BRM billing record the difference between the old and re-rated balance impacts for each EDR.

The rerating output file contains all events for which the price has changed as a result of rerating and the change in the balance impact.

The output module sends the output file with the updated impact balances to the Rerated Event Loader (RREL) which loads them into the BRM database.

Figure 5-2 shows the rerating pipeline.

**Figure 5-2 Rerating Pipeline**

# About Configuring the Rerating Pipeline

When you configure the rerating pipeline, you configure the same modules in the registry as for the normal rating pipeline. The only difference is that you do not configure the FCT_DuplicateCheck and the FCT_CallAssembling modules.

For the rerating pipeline, you use special backout input and output grammar files, and you create separate input and output directories.

For a sample registry for the rerating pipeline, see the **rerating.reg** file.

This sample shows the input configuration for a rerating pipeline. It uses a special backout grammar file, and a separate input directory.

```
Input
  {
    UnitsPerTransaction = 1
      InputModule
      {
        ModuleName  = INP_GenericStream
        Module
        {
          DefaultOutput = RerateOutput
          Grammar = ./formatDesc/Formats/Solution42/SOL42_V630_REL_InGrammar_BACKOUT.dsc
          InputStream
          {
            ModuleName = EXT_InFileManager
            Module
            {
              InputDirEmptyTimeout = 10
              InputPath   = ./samples/wireless/data/backout/in
              InputPrefix = test.
              InputSuffix = .edr
              DonePath    = ./samples/wireless/data/backout/done
              DonePrefix  = test.
              DoneSuffix  = .done
              ErrorPath   = ./samples/wireless/data/backout/err
              ErrorPrefix = test.
              ErrorSuffix = .err
              TempPrefix  = tmp.
              Replace     = TRUE
            }
          } # end of InputStream
        }
      } # end of InputModule
    } # end of Input
```

This sample shows the output for telephony EDRs in a rerating pipeline. It uses a special backout output grammar file and a separate output directory.

```
TELOutput
   {
     ModuleName = OUT_GenericStream
     Module
     {
       Grammar = ./formatDesc/Formats/Solution42/SOL42_V630_REL_OutGrammar.dsc
       DeleteEmptyStream = True
      OutputStream
       {
         ModuleName = EXT_OutFileManager
         Module
```

```
        {
          OutputPath = ./samples/wireless/data/rerate/telout
          OutputPrefix = test
          OutputSuffix = .out
          TempPrefix   = .
         TempDataPath    = ./samples/wireless/data/rerate/telout
          TempDataPrefix  = tel.tmp.
          TempDataSuffix  = .data
         Replace = TRUE
        }
      }
    }
  } # end of TELOutput
```

> **Note:**
>
> To ensure output file integrity, specify a unique combination of OutputPath, OutputSuffix, and OutputPrefix values for each output stream defined in the registry.

## About Starting the Rerating Pipeline

Before you start the rerating pipeline, ensure that the back-out pipeline has processed all EDRs that were extracted by the Event Extraction Tool. Also ensure that all other rating pipelines are stopped.

To start the rerating pipeline, use the event handler to start the rerating pipeline when the input directory is empty. To do so, configure the corresponding section in the registry. The EVT_INPUT_DIR_EMPTY event is triggered by the EXT_InFileManager when the input directory is emptied. In this sample, the event starts the rerating pipeline.

```
EventHandler
{
  ModuleName = EVT
  Module
  {
    Events
    ifw.Pipelines.ALL_BCKOUT.Input.InputModule.Module.InputStream.Module
    {
      EVT_INPUT_DIR_EMPTY = ./start_rerate.sh
    }
  }

  Buffer
  {
    Size = 100
  }
 }
}
```

To define when to send EVT_INPUT_DIR_EMPTY event, use the EXT_InFileManager **InputDirEmptyTimeout** registry entry.

## About Stopping the Rerating Pipeline

You stop the rerating pipeline manually by sending a semaphore:

```
ifw.Pipelines.RERATE.Active = FALSE
```

6

# About Comprehensive Rerating Using pin_rerate

Learn about rerating using the Oracle Communications Billing and Revenue Management (BRM) **pin_rerate** utility to rerate real-time-rated events and pipeline-batch-rated events.

> **Note:**
>
> This documentation does not apply to rerating only pipeline-batch-rated events by using a batch rerating pipeline.

Topics in this document:



## About Comprehensive Rerating

The **pin_rerate** utility provides a comprehensive rerating solution: You can rerate both real-time-rated and pipeline-batch-rated events concurrently or, if you do not use Pipeline Manager for batch rating, you can rerate only real-time-rated events. You can select accounts and events for rerating based on any event criteria. And you can configure BRM to automatically rerate certain accounts and events when you run **pin_rerate**.

The **pin_rerate** utility controls the rerating workflow. It follows a different rerating process depending on whether batch pipeline rating is enabled. See:

- About Rerating Pipeline and Real-Time Events Concurrently
- About Rerating Events When You Do Not Use Pipeline Batch Rating

### About Rerate Jobs

A rerate job is a set of objects in the BRM database used to track the accounts selected for rerating and the status of the rerating process. Rerate jobs are created for all accounts to be rerated by **pin_rerate**. (For more information about rerate job objects, see "How BRM Creates Rerate Jobs".)

Rerate jobs are typically associated with a batch of accounts, but can also be associated with a single account.

Rerate jobs are created manually when you use **pin_rerate** to select accounts for rerating. Rerate jobs are created automatically when certain events occur, such as when backdated events or rate changes trigger automatic rerating.

When you use **pin_rerate** to select accounts for rerating, the utility assigns accounts to each rerate job and creates 2 rerate jobs per transaction. This enables **pin_rerate** to more easily roll back a transaction if an error occurs.

You can configure the number of accounts assigned to each rerate job and the number of jobs to create per transaction.

# About Rerating Real-Time-Rated and Pipeline-Rated Events Concurrently

When you use **pin_rerate** to rerate both real-time-rated and pipeline-batch-rated events concurrently, the following tasks are performed:

1. The utility retrieves the accounts for rerating from the BRM database based on search criteria that you specify.
2. BRM extracts the events associated with those accounts and then backs out the original event balance impacts.
3. BRM determines whether the original event was rated in real time or by the batch rating pipeline:
   a. It sends events previously rated in real time to the real-time rating opcodes for rating and discounting.
   b. It sends events previously rated by a batch pipeline to the real-time rerating pipeline for rating and discounting. See "About the Real-Time Rerating Pipeline".
4. BRM records the rerated events in the BRM database.

## About the Real-Time Rerating Pipeline

The **pin_rerate** utility sends events previously rated by the batch rating pipeline to a real-time rerating pipeline to be rerated.

The **pin_rerate** utility routes the pipeline-rated events to the Connection Manager (CM). The CM sends the pipeline-rated events in the form of an flist to the NET_EM pipeline module. The NET_EM module transfers the event to the real-time rerating pipeline for rerating.

The real-time rating opcodes add to the flist all the enrichment data needed by the real-time rerating pipeline to rate the event. For example, if the real-time rating opcodes determine that a discount should be applied, it adds the discount information to the flist.

> **Note:**
>
> Real-time rerating process does not check credit limits when applying discounts. To configure rerating to check credit limits, create a custom iScript that sets the DETAIL.CREDIT_LIMIT_CHECK EDR field to **1**.

Similar to a batch rerating pipeline, a real-time rerating pipeline performs both rating and discounting functions.

To rerate events, the real-time rerating pipeline gets the new pricing information from the Pipeline Manager database. Certain configuration data, such as currency and noncurrency balance element information, are obtained from the BRM database.

The real-time rerating pipeline performs the following tasks:

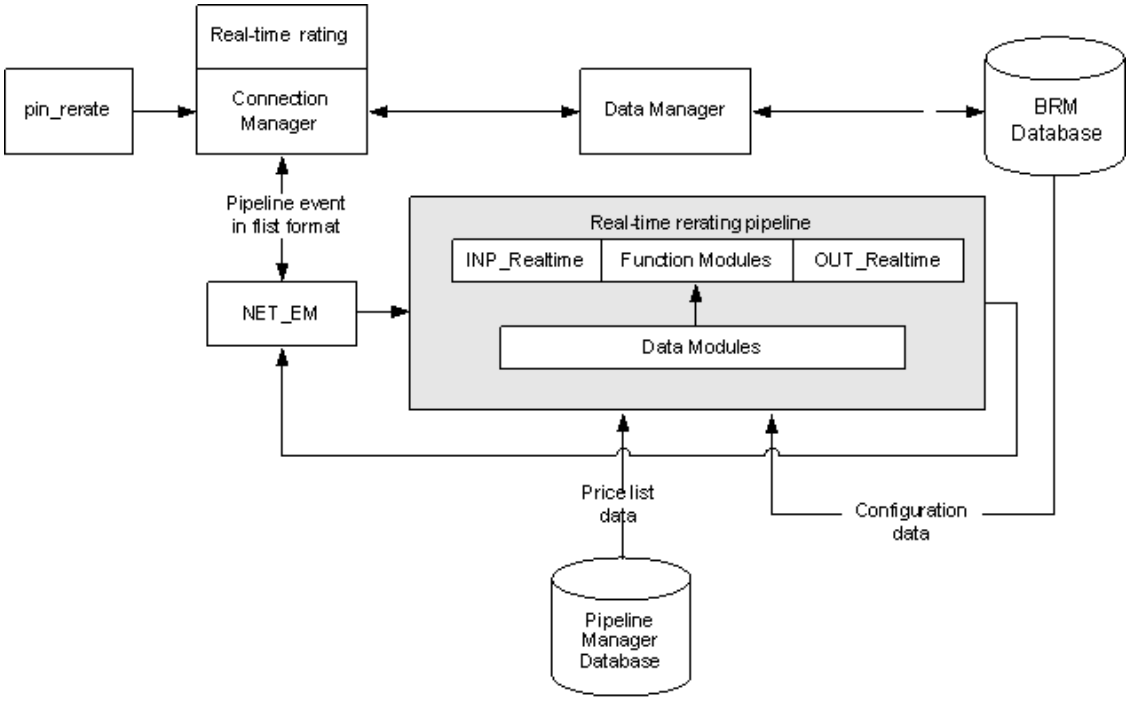
1. The INP_Realtime input module converts the flist received from the NET_EM module to event data record (EDR) format and creates an EDR container.

> **Note:**
>
> You can also create and run an iScript to manipulate data in the EDR container before it is sent to the pipeline modules.

2. The pipeline function modules calculate the new balance impacts by using the new pricing and discounting information and then adds the balance impacts to the EDR container.
3. The OUT_Realtime module converts the EDR back into flist format and sends the flist to NET_EM.
4. NET_EM sends the flist with the new balance impacts back to the CM.
5. BRM updates the account's balances and records the event in the BRM database.

Figure 6-1 shows the data flow for rerating events by using the real-time rerating pipeline.

**Figure 6-1 Real-Time Rerating Pipeline Data Flow**



## About Transaction Management for the Real-Time Rerating Pipeline

The real-time rerating pipeline does not use the Transaction Manager (TAM) module. Instead, transaction handling is provided by the CM. When a rerating transaction fails, the CM rolls back the transaction and restores all the account balances in the BRM database. For this reason, function modules and iScripts used by the real-time rerating pipeline are stateless.

> **Note:**
>
> If you use iScripts for custom processing, ensure that the iScripts are stateless.

# How pin_rerate and the Batch-Rating Pipeline Synchronize Processes

While pipeline-rated events are being rerated, the batch pipeline temporarily suspends new EDRs for the accounts that are currently being rerated. When rerating is complete, account balance data in Pipeline Manager is synchronized with the account balance data in the BRM database, and the batch pipeline resumes processing the suspended EDRs using the updated data.

The batch pipeline and **pin_rerate** use the Oracle DM and BRM standard recycling to synchronize processes.

When you run **pin_rerate** for concurrent rerating, the following actions are taken to synchronize **pin_rerate** and batch pipeline processes:

1. The **pin_rerate** utility retrieves the accounts to be rerated and creates the rerate jobs. The rerate jobs are queued until they are processed.
2. **pin_rerate** sends a business event through the Oracle DM to notify the batch pipeline that rerating for the selected accounts is about to begin. When it receives the notification, the batch pipeline suspends rating incoming CDRs for the accounts and sends **pin_rerate** a corresponding acknowledgment event.
3. **pin_rerate** rerates the accounts in the rerate job and BRM records the new balance impacts of the rerated events in the BRM database.
4. **pin_rerate** sends a business event through the Oracle DM to notify the batch pipeline that rerating is complete for the selected accounts. The batch pipeline synchronizes the account data in pipeline memory with the account data in the BRM database and resumes processing EDRs. The batch pipeline sends **pin_rerate** a corresponding acknowledgment event.
5. **pin_rerate** recycles the EDRs suspended during the rerating process by calling the standard recycling opcode.

After successful completion of each step, **pin_rerate** updates the rerate job status so that the next step in the rerate workflow can be run.

Rerate jobs that are processed during concurrent rerating can have the statuses shown in Table 6-1.

**Table 6-1 Status for Rerate Jobs**

| Status | Description |
|---|---|
| NEW | The initial status of a rerate job. |
| WAITING_ACCOUNT_LOCKED | The status after **pin_rerate** has notified the batch pipeline to suspend EDRs for accounts in the rerate job. |
| ACCOUNT_LOCKED | The status after **pin_rerate** has received acknowledgment from the batch pipeline. |
| RERATED | The status after the events associated with the accounts in the rerate jobs have been rerated. |
| READY_FOR_RECYCLE | The status after account data has been synchronized between the BRM database and Pipeline Manager and when the batch pipeline has resumed processing EDRs for the accounts in the rerate job. |
| COMPLETE | The status when rerating is completed for *all* the accounts in the rerate job. |

## About Suspending and Recycling EDRs during the Rerating Process

To suspend EDRs for Accounts that are being rerated, the pipeline DAT_AccountBatch module sets a rerate error code and a recycle key value in the EDR. The error code causes the EDR to be suspended by BRM standard recycling. The recycle key is used to select the EDRs to be recycled.

You must load suspended EDRs into the BRM database by running Suspended Event (SE) Loader. You typically schedule SE Loader to run automatically when you set up standard recycling.

Suspended EDRs are stored in the database until they are recycled. To recycle the EDRs that were suspended during the rerating process, you run **pin_rerate** with the **-process recycle** parameter after rerating is completed.

# Procedure for Concurrent Rerating of Real-Time-Rated and Pipeline-Rated Events

To rerate both real-time-rated events and pipeline-batch-rated events concurrently, you perform the following steps:

1. Run **pin_rerate** to select the accounts for rerating and create rerate jobs.

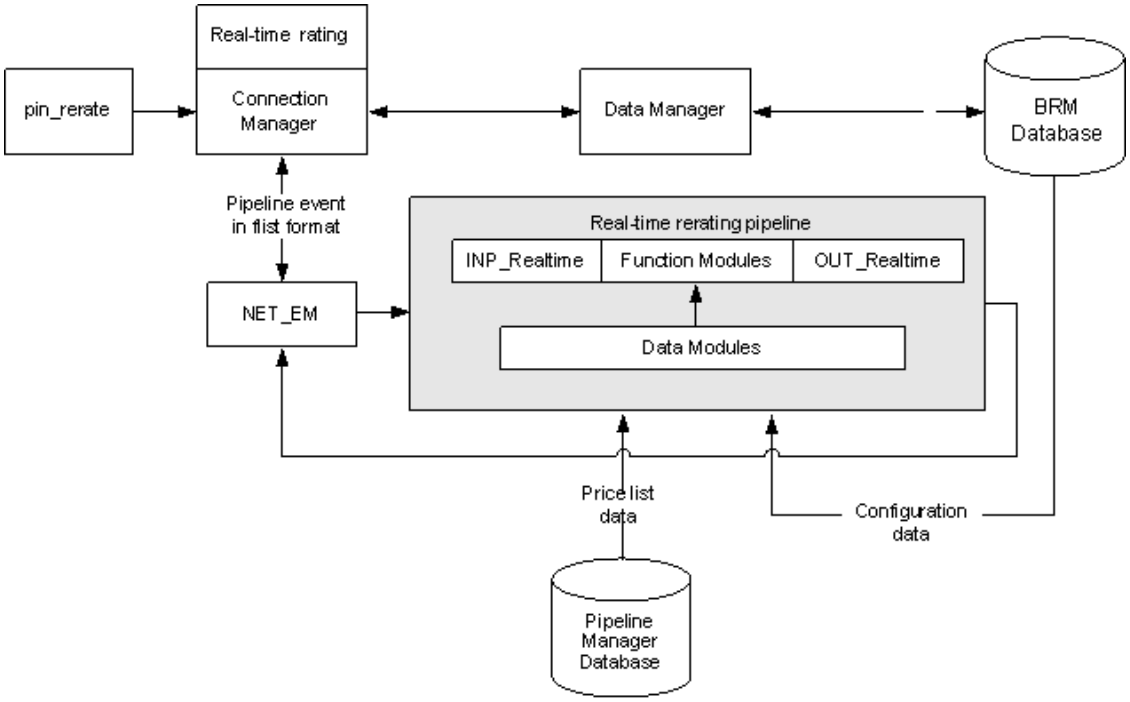
   > **Note:**
   >
   > Rerate jobs can also be created automatically through BRM automatic rerating or trigger-dependent rerating.

2. Run a series of **pin_rerate** commands to rerate the events associated with the accounts in the rerate job.

3. Run **pin_rerate** to recycle the EDRs associated with the accounts in the rerate job that were suspended during rerating.

Figure 6-2 shows the rerating commands, process flow, and job status transitions when you perform concurrent rerating.

**Figure 6-2 Concurrent Rerating Process Flow**




# About Rerating Events When You Do Not Use Pipeline Batch Rating

If you do not use Pipeline Manager for batch rating, you can use **pin_rerate** to rerate only real-time-rated events.

> **Note:**
>
> Real-time-rated events are events that are rated by the BRM rating opcodes. Real-time-rated events include events that are loaded in batches by Universal Event (UE) Loader.

To rerate only real-time-rated events, you must configure the **BatchRatingPipeline** business parameter so that **pin_rerate** does not attempt to communicate with a batch pipeline, which will cause rerating to fail.

When rerating only real-time-rated events, the following actions are taken when you run **pin_rerate**:

1. The **pin_rerate** utility selects the accounts to be rerated from the BRM database based on the specified search criteria and creates rerate jobs.

> **Note:**
>
> Rerate jobs can also be created automatically through BRM automatic rerating or trigger-dependent rerating.

2. **pin_rerate** invokes rerating of the selected accounts in the rerate jobs.
3. BRM extracts the events associated with the selected accounts from the BRM database and does the following:
   a. Rerates the real-time-rated events.
   b. Records the new balance impacts of the rerated events in the BRM database.

After the successful completion of each step, **pin_rerate** updates the rerate job status so that the next step in the rerate workflow can be run.

Rerate jobs that are processed when rerating only real-time-rated events can have the statuses shown in Table 6-2.

**Table 6-2 Status for Real-Time Rerate Events**

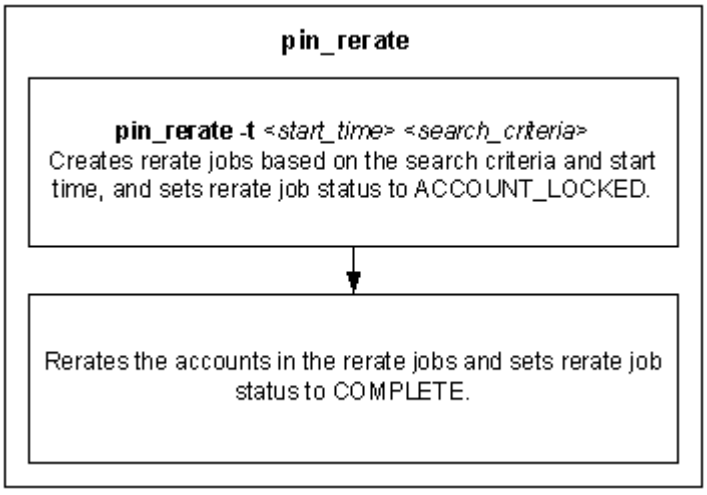
| Status | Description |
|---|---|
| ACCOUNT_LOCKED | The initial status of rerate jobs. |
| COMPLETE | The status when rerating has completed for *all* the accounts in the rerate job. |

## Procedure for Rerating Only Real-Time-Rated Events

To rerate only real-time-rated events when you do not use Pipeline Manager for batch rating, you select the accounts to rerate and rerate the selected accounts using a single **pin_rerate** command.

Figure 6-3 shows the rerating command, process flow, and job status transitions when rerating only real-time-rated events.

**Figure 6-3 Rerating Only Real-Time Events**



# How Failed Rerate Jobs Are Processed

The accounts in a rerate job are typically processed individually in separate rerate operations. When rerating fails for one or more accounts in a rerate job, **pin_rerate** sets the status of the accounts in the rerate job batch to FAILED. When rerating process is complete for the rerate job, **pin_rerate** creates a new rerate job consisting of only the accounts that failed. The new rerate job is processed the next time **pin_rerate** is run.

If an account selected for rerating is associated with a subscription service that was transferred during the period for which rerating is specified, then the account to which the service was transferred is included in the rerate job and all those accounts are rerated concurrently in a single rerate operation. When rerating fails for one of these accounts, then rerating fails for all accounts in the rerate request. In this case, **pin_rerate** creates a new rerate job containing all the accounts in the rerate request.

For example, subscription service X is originally owned by Account A and transferred to Account B on June 15. Later in the month, it is transferred from Account B to Account C. Rerating of Account A from June 1 also results in rerating of accounts B and C. Accounts A, B, and C are grouped together in a single rerate request. If rerating fails for any of these accounts, **pin_rerate** creates a new rerate job consisting of all three accounts in a single rerate request. The accounts are rerated again the next time **pin_rerate** is run.

# About Automatic Rerating

When certain events occur in the BRM system (such as backdated purchase events), customer accounts require rerating so they are billed accurately. In automatic rerating, BRM automatically creates rerate jobs for these events.

> **Note:**
>
> Automatic rerating *does not* immediately rerate accounts when the rerate job is created. You must still run the **pin_rerate** utility.

Accounts and events associated with rerate jobs that are created automatically are rerated the next time you run **pin_rerate** to process rerate jobs: you do not need to first select the accounts and events by specifying **pin_rerate** selection criteria.

You enable automatic rerating by configuring event notification to call the PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST policy opcode when events occur that require rerating.

Automatic rerating uses rerate reason codes. A rerate reason code is passed in to PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST, which uses the code to help determine whether rerating is required for an event.

BRM creates rerate jobs for automatic rerating as follows:

1. An event occurs that requires accounts to be rerated.
2. The opcode that triggers rerating when that particular event occurs creates a notification event.

   Table 6-3 summarizes the types of notification events used to trigger automatic rerating and the rerating scenario to which each event applies.

**Table 6-3 Types of Notifications for Automatic Rerating**

| Notification Events for Automatic Rerating | Rerating Scenario |
|---|---|
| **/event/notification/auto_rerate** | Automatic rerating of backdated events. |
| **/event/notification/rate_change** | Automatic rerating for rate changes. |
| **/event/notification/rollover_correction/rerate** | Automatic rerating of rollover corrections due to delayed events. |
| An event by which you want rerating to be triggered | Automatic rerating for your custom rerating requirements. |

3. The event notification mechanism calls PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST.
4. PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST takes as input the event type and the rerate reason code associated with the event and analyzes the event to determine if rerating is required.

> **Note:**
>
> BRM reserves the range of 100 to 120 for automatic rerating reason codes.

5. If rerating is required, PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST calls the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode along with the appropriate rerating criteria to create a rerate job.

PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST calls PCM_OP_SUBSCRIPTION_INSERT_RERATE_REQUEST only when the reason code passed in is one of those reserved for automatic rerating (reason codes between 100 and 120). By default, if a different reason code is passed in, the opcode does nothing. To create rerate jobs for reason codes other than those reserved for automatic rerating, you must customize PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST.

By default, PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST passes a reason code of **0** (no rerate reason) to PCM_OP_SUBSCRIPTION_INSERT_RERATE_REQUEST. To rerate accounts separately based on a rerate reason code, customize PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST to pass the rerate reason code it receives (or a different rerate reason code). The rerate reason code is stored in the rerate job object, and you can select rerate jobs based on the specific rerate reason code when you run **pin_rerate** to process rerate jobs.

> **Note:**
>
> You can also assign reason codes when you manually create rerate jobs by using **pin_rerate**.

You can also customize PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST in other ways. For example, you can change whether rerate jobs are created for specific events: such as when you want to rerate events for only a few (rather than all) automatic rerating scenarios.

# How Rerating Affects Account Migration

When an account is selected for rerating, the account cannot be migrated to another database until rerating is complete. Otherwise, the account is not rerated. For this reason, the Account Migration Manager (AMM) does not allow migration of an account to another database if the account is being rerated by **pin_rerate**.

The AMM does not migrate an account if the account is in a rerate job and the rerate job status is one of the following:

- WAITING_ACCOUNT_LOCKED
- ACCOUNT_LOCKED
- RERATED
- READY_FOR_RECYCLE

However, the account is migrated if the rerate job status is NEW. In this case, the AMM deletes the account from the rerate job in the source database and creates a new rerate job with the account in the destination database.

> **Note:**
>
> After the account migration is complete, you must run **pin_rerate** in the destination database to rerate the account.

# Managing Comprehensive Rerating with Custom Applications

BRM uses the following opcodes for comprehensive rerating with **pin_rerate**:

- PCM_OP_SUBSCRIPTION_RERATE_REBILL. See "How Comprehensive Rerating Works".
- PCM_OP_RERATE_INSERT_RERATE_REQUEST. See "How BRM Creates Rerate Jobs".
- PCM_OP_SUBSCRIPTION_PREP_RATE_CHANGE.
- PCM_OP_SUBSCRIPTION_RATE_CHANGE.
- PCM_OP_SUBSCRIPTION_POL_SPEC_RERATE.

## How Comprehensive Rerating Works

To rerate events, use PCM_OP_SUBSCRIPTION_RERATE_REBILL.

This opcode rerates the events for accounts identified by the **pin_rerate** utility, rerating one account at a time. The rerating start time is specified in the input flist. This opcode calls other opcodes to perform rerating functions.

PCM_OP_SUBSCRIPTION_RERATE_REBILL generates either a shadow event or an adjustment event to apply the results of rerating to the account balance in the current billing cycle:

- If the event is unbilled, a shadow event is created, which is an adjustment event that is applied to the bill item instead of the adjustment item.
- If the event is billed, or if the event is unbilled but already posted to the G/L, an adjustment event is created and applied to the adjustment item.

PCM_OP_SUBSCRIPTION_RERATE_REBILL does not rerate, but only reapplies the balance impacts of the original event for these event types: adjustment, payment, writeoff, dispute, settlement, refund, charge, item transfer, cycle fold, cycle tax, reversal, and pre-rated.

If there is no balance impact associated with an event or if there is no rated quantity on the original event, the event is not rerated.

The input to PCM_OP_SUBSCRIPTION_RERATE_REBILL includes:

- The POID of the account to rerate.
- The time from which to start rerating. Events occurring from this start time are rerated.
- Flags that modify rerating behavior. See "Flags Used for Rerating".
- The session object. When PCM_OP_SUBSCRIPTION_RERATE_REBILL is invoked by the **pin_rerate** utility, the session object passed is the rerating audit object (**/event/control_rerate**). This object is used to generate rerating reports.

PCM_OP_SUBSCRIPTION_RERATE_REBILL does the following:

1. Closes the billing cycle, if due.
2. Retrieves all events for the account from the start date specified.
3. Determines the balance for each balance element in the account when rerating starts. This is the current balance minus the cumulative total of all balance impacts for the balance element from the rerating start time until the current time.
4. For each event:
   - If the effective time is passed in the PIN_FLD_START_T field, no balance impact is required and the event is not rerated.
   - Checks the **/data/ledger_report** object to determine whether the G/L for the event has been posted.
   - Calls the PCM_OP_ACT_USAGE opcode to determine the difference between the original rated event and the rerated event.

     > **Note:**
     >
     > PCM_OP_ACT_USAGE does not check credit limits when rerating events.

   - Calls PCM_OP_ACT_USAGE again to apply the results of rerating to the account balance. If the event is sponsored, the balance impacts are applied to the sponsoring account.
   - Updates the RERATE_OBJ field for the event being rerated with the POID of the new rerated event.

The account balance is updated after rerating each event so that each subsequent event is rerated based on the most current balance.

Some rerated events may be associated with a charge offer, discount offer, or balance element balance whose validity period starts on first usage. If the event that triggered the validity period is not the actual first usage event, PCM_OP_SUBSCRIPTION_RERATE_REBILL backs out

the validity period of the charge offer, discount offer, or balance element balance and resets it based on the event time of the actual first-usage event.

When override pricing is passed to PCM_OP_SUBSCRIPTION_RERATE_REBILL, it creates a **/rerate_session/override_info** object to capture the override pricing information used during rerating.

## Flags Used for Rerating

The PIN_FLD_FLAGS field specifies whether to rerate events in order based on the event creation time or the event end time. The end time is the default.

The PIN_FLD_RERATE_FLAGS field can take two flags:

- A flag that specifies back-out-only rerating. When this flag is set, the balance impacts of the events selected for rerating are backed out, but the events are not rerated.
- A flag that specifies selective rerating of specific events. When this flag is set, the account selection criteria is also applied to the account's events and only events that meet the selection criteria are rerated.

> **Note:**
>
> – Do not use selective rerating if your rating scheme includes credit limits or balance element-based tiers. These schemes require that all events related to an account are rerated to assure accurate rerating.
>
> – Do not use selective rerating if deferred taxation is used for taxing events during rerating.
>
> – Use caution when specifying selective rerating. Account balances can be impacted by different types of events and the order of the balance impacts is important to accurately apply discounts and consume balance elements. It is typically safer to rerate all of an account's events.

The selection criteria for selective rerating and back-out-only rerating must be set in the PIN_FLD_ARGS array field in the input flist.

## Return Values for Rerating

If the CALC_ONLY flag is set or if PCM_OP_FLAG_READ_RESULT is passed in the input flist, PCM_OP_SUBSCRIPTION_RERATE_REBILL returns all rerating details in the PIN_FLD_RESULTS array without modifying the account balances. Otherwise, the account balances are updated and only the POID of the rerated event is returned.

If PCM_OP_SUBSCRIPTION_RERATE_REBILL attempts to rerate an event that is already being rerated, an error is returned.

# How BRM Creates Rerate Jobs

A rerate job is a pair of **/job/rerate** and **/job/rerate** objects in the BRM database. There is a one-to-one relationship between **/job/rerate** objects and **/job_batch/rerate** objects and both are created in the same transaction when one or more accounts are selected for rerating.

You create rerate jobs manually by using the **pin_rerate** utility. BRM creates rerate jobs automatically by using the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode. This opcode is called to create rerate jobs by the following:

- The **pin_rerate** utility.
- BRM's automatic-rerating mechanism.

  BRM supports the automatic rerating of certain types of events. For more information, see "About Automatic Rerating".
- Your trigger-dependent rerating configuration.

  BRM enables you to rerate events automatically based on your own business requirements.
- Out-of-order rerating for pipeline rated events.

  BRM supports the automatic rerating of events that are rated out of order in the batch rating pipeline.

PCM_OP_RERATE_INSERT_RERATE_REQUEST receives the following information in the input flist:

- The POIDs of the accounts to be rerated (required).
- The time from which events must be rerated for the accounts (required).
- Accounts related to the accounts to be rerated (for example, an account that used the account's service before a line transfer) along with the start time for each account.

  > **Note:**
  >
  > If more than one account is included in the rerate job, the same rerate start time, selection criteria, and price overrides apply to all the accounts.

- Your rerate reason for trigger-dependent rerating (the default rerate reason passed in for BRM automatic rerating scenarios is **0**).
- Your selection criteria for trigger-dependent rerating to identify the events that must be rerated for the accounts.
- A list of charge offers, discount offers, or bundles to override the subscriber's existing pricing objects during rerating.

To create rerate jobs, PCM_OP_RERATE_INSERT_RERATE_REQUEST does the following:

1. Checks for duplicate rerate job requests.

   For more information, see "How BRM Handles Duplicate Rerate Jobs".
2. Calls PCM_OP_RERATE_CREATE_JOB to create the **/job/rerate** and **/job_bacth/rerate** objects.

## How BRM Handles Duplicate Rerate Jobs

The PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode checks for existing jobs when a new rerate request is made; this opcode compares the data for each top-level account in the rerate request to the data in the existing jobs. To avoid duplicate rerate jobs, accounts are either eliminated from or updated in the rerate request or the existing jobs.

After the duplicate detection process is complete for each top-level account in the rerate request, PCM_OP_RERATE_INSERT_RERATE_REQUEST calls PCM_OP_RERATE_CREATE_JOB to create a rerate job with the resulting contents of the rerate request. PCM_OP_RERATE_CREATE_JOB then performs checks on the sub-accounts in the rerate request to check for duplicate jobs for line transfers before creating the rerate job.

## Detecting Duplicate Rerate Requests

When the rerate reason code and rerate selection criteria for a rerate request match those of one or more rerate jobs, the request is considered a possible duplicate. When the rerate reason code and rerate selection criteria do not match, the rerate request is not a duplicate, and a new rerate job is created.

## Avoiding Duplication of Rerate Jobs

To avoid duplicate rerate jobs for an account, PCM_OP_RERATE_INSERT_RERATE_REQUEST compares the following values for duplicate rerate job requests:

- The rerate start times. BRM always uses the earlier start time to ensure that all events are rated.
- The price overrides (if any). BRM uses the latest price overrides for an account because it is assumed to be the most updated.

**When price overrides match**

When the price overrides match, PCM_OP_RERATE_INSERT_RERATE_REQUEST does the following for each top-level account:

- If the rerate start time of the rerate request is later than or equal to the start time of the existing job, does the following:
  - If there is only this account in the rerate request, deletes the request. No rerate job is needed.
  - If there are many accounts in the new request, removes only that account from the new request.
- If the rerate start time of the rerate request is earlier than the start time of the existing job, does the following:
  - If the existing job contains only this account, updates the existing job to use the earlier start time and removes the account from the rerate request.
  - If the existing job contains other accounts, removes the account from the existing job, keeping the account in the rerate request.

**When price overrides do not match**

When the price overrides do not match, PCM_OP_RERATE_INSERT_RERATE_REQUEST does the following for each top-level account:

- If the rerate start time of the rerate request is the same or earlier than the start time of the existing job, does the following:
  - If there is only one account in the existing job, deletes the job.
  - If there are many accounts in the existing job, keeps the account in the rerate request so the new rerate job start time and price overrides are used. Removes the account from the existing job.

- If the rerate start time of the rerate request is later than the start time of the existing job, does the following:
  - If there is only one account in the existing job, deletes the job.
  - If there are many accounts in the existing job, keeps the account in the rerate request so the new price overrides are used. Updates the rerate request start time to use the existing job's start time. Removes the account from the existing job.

Table 6-4 summarizes how PCM_OP_RERATE_INSERT_RERATE_REQUEST handles duplicate rerate job requests.

**Table 6-4 Request Handling by PCM_OP_RERATE_INSERT_RERATE_REQUEST**

| If the Start Time of the New Request Is: | And the Price Overrides: | Perform This Action: |
|---|---|---|
| Equal to or later than that of the existing job | Match | If there is only this account in the rerate request, delete the request. No rerate job is needed.<br>If there are many accounts in the rerate request, remove only that account from the rerate request. |
| Earlier than that of the existing job | Match | If there is only one account in the existing job:<br>• Update the existing job to use the earlier start time.<br>• Remove the account from the rerate request.<br>If there are many accounts in the existing job:<br>• Keep the account in the rerate request.<br>• Remove the account from the existing job. |
| Later than that of the existing job | Do not match | If there is only one account in the existing job, delete the job.<br>If there are many accounts in the existing job:<br>• Keep the account in the rerate request so the new price overrides are used.<br>• Update the rerate request job start time to use the existing job's start time.<br>• Remove the account from the existing job. |
| Equal to or earlier than that of the existing job | Do not match | If there is only one account in the existing job, delete the existing job.<br>If there are many accounts in the existing job:<br>• Keep the account in the rerate request so the rerate request start time and price overrides are used.<br>• Remove the account from the existing job. |

For more information on how BRM creates rerate jobs, see "How BRM Creates Rerate Jobs".

## Rerating Cycle Fees

To rerate cycle fees, use the PCM_OP_SUBSCRIPTION_RATE_CHANGE opcode. This opcode uses the event notification mechanism to trigger the creation of rerating requests when there is a cycle-forward or cycle-forward-arrears event rate change in the middle of the current cycle.

> **Note:**
>
> Rerating is not triggered for cycle-arrears rate changes or rate changes in future cycles.

When you run the **pin_rate_change** utility after a charge pricing change, the utility calls this opcode and provides details about the accounts and charge offers affected by the rate change.

This opcode returns a notification event of type **/event/notification/rate_change** for each account picked up by the **pin_rate_change** utility. Depending on how automatic rerating is configured, the notification event triggers the creation of rerating requests.

7

# Configuring Comprehensive Rerating

Learn how to configure your Oracle Communications Billing and Revenue Management (BRM) system for comprehensive rerating using **pin_rerate**.

> **Note:**
>
> This documentation does not apply to rerating only pipeline-batch-rated events by using a batch rerating pipeline.

Topics in this document:



## About Configuring Comprehensive Rerating

How you configure rerating depends on whether you rerate real-time-rated and pipeline-batch-rated events concurrently or, if you do not use Pipeline Manager for batch rating, rerate only real-time-rated events. See one of the following sections:

- Configuring Concurrent Rerating of Pipeline-Rated and Real-Time-Rated Events
- Configuring Rerating When You Do Not Use a Batch Rating Pipeline

You perform the following configurations for both concurrent rerating and real-time-event only rerating:

- Specifying Whether the Batch Rating Pipeline Is Enabled
- (Optional) "Setting the Rerating Event Cache Size (Fetch Size)"
- (Optional) "Configuring the Number of Accounts Per Job and Number of Jobs per Transaction"

You can also configure BRM to automatically create rerate jobs when the following events occur:

- When rates are changed.
- When rollover corrections are made due to delayed events.
- When certain events, such as purchase and cancellation events, are backdated. See "About Automatic Rerating of Backdated Events".
- When pipeline batch-rated events are rated out of order. See "About Automatic Rerating of Out-of-Order Events".
- When events trigger rerating based on your custom rerating requirements. See "About Trigger-Dependent Rerating".

# Configuring Concurrent Rerating of Pipeline-Rated and Real-Time-Rated Events

> **Note:**
>
> Before you can configure concurrent rerating, you must have installed and configured Pipeline Manager and the Oracle Data Manager (DM).

To configure concurrent rerating of both pipeline-rated and real-time-rated events, perform these tasks:

- Configure a real-time rerating pipeline. See "Configuring a Real-Time Rerating Pipeline".
- Configure **pin_rerate** and the batch rating pipeline. See "Configuring the Batch Rating Pipeline and pin_rerate to Synchronize Processing".
- (Optional) Configure the rerating event cache size. See "Setting the Rerating Event Cache Size (Fetch Size)".
- (Optional) Configure parameters for creating rerate jobs. See "Configuring the Number of Accounts Per Job and Number of Jobs per Transaction".

## Configuring a Real-Time Rerating Pipeline

A real-time rerating pipeline is configured similarly to a batch rerating pipeline, but with fewer preprocessing and post-rating modules. This is because most of the enrichment data is provided in the input flist received from the CM.

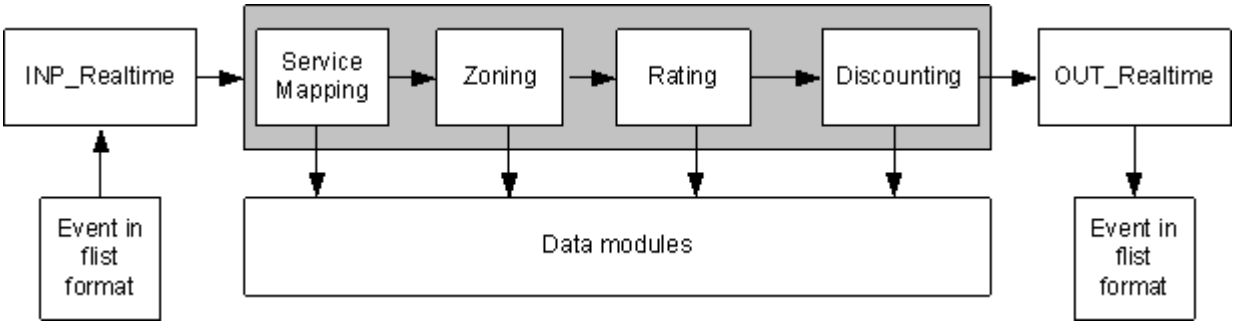
You typically create a separate instance of Pipeline Manager for real-time rerating. The default registry file is *BRM_home***/conf/wirelessRealtime.reg**.

To configure a real-time rerating pipeline, perform the following tasks:

- (Optional) "Configuring Multiple Real-Time Rerating Pipelines"
- Configuring the Real-Time Rerating Data Pool
- Configuring the Modules in the Real-Time Rerating Pipeline
- Adding Real-Time Rerating Pipeline Data to the IFW_PIPELINE Table

Figure 2-2 shows the real-time rerating pipeline architecture:

**Figure 7-1 Real-Time Rerating Pipeline Architecture**




## Configuring Multiple Real-Time Rerating Pipelines

To improve performance or scalability, you configure multiple instances of real-time rerating pipelines. For example, you can increase performance by configuring multiple pipelines to process rerate requests in parallel or by configuring multiple real-time rerating pipelines to process different rerate requests: for example, by configuring one pipeline that rerates only GPRS events and another that rerates only GSM events.

## Configuring the Real-Time Rerating Data Pool

Configure the following data modules:

- DAT_Zone
- Rateplan
- DAT_Calendar
- DAT_TimeModel
- DAT_PriceModel
- Dayrate
- DAT_Currency
- DAT_USC_Map

## Configuring the Modules in the Real-Time Rerating Pipeline

Configure the following modules in the real-time rerating pipelines:

- Configure the INP_Realtime input module. Specify the flist-to-EDR mapping file used for rerating in the **OpcodeMapping** registry entry. For example:

  **OpcodeMapping = ./formatDesc/Formats/Realtime/rate_event.xml**

- Configure the NET_EM module:
  - Configure NET_EM to manage data between the CM and Pipeline Manager and configure the CM to send rerate request to the NET_EM module.
  - Configure NET_EM to route rerate requests. See "Configuring NET_EM to Route Rerate Requests Based on the Event Field Value".
- Configure the output module to add any customizations to the output flist.
- Configure these function modules:

- FCT_ServiceCodeMap
- FCT_CustomerRating
- FCT_PreRating
- FCT_IRules
- FCT_USC_Map
- FCT_RSC_Map
- FCT_MainRating
- FCT_Dayrate
- FCT_RateAdjust
- FCT_Rounding
- FCT_DiscountAnalysis
- FCT_Discount

## Configuring NET_EM to Route Rerate Requests Based on the Event Field Value

To configure NET_EM to route rerate requests to multiple real-time rerating pipelines based on the type of event, you set the **FieldName** and **FieldValue** NET_EM module registry entries.

> **Note:**
>
> If you use event routing based on the event field value, ensure that the input events contain the expected field name and field values specified in the NET_EM module registry. Otherwise, NET_EM will not be able to route the events.

By using the "**.**" notation, you can specify a field at any level in the input event flist to be used to route the event. For example, this substruct and field:

```
PIN_FLD_EVENT
     PIN_FLD_POID
```

is represented like this:

```
PIN_FLD_EVENT.PIN_FLD_POID
```

In the NET_EM registry below, if PIN_FLD_EVENT.PIN_FLD_POID is a GSM event, the rerate request is routed to any one of the two instances of the GSM rerating pipeline (RealtimeReratingGSM). If the event is a GPRS event, the rerate request is routed to any one of the two instances of the GPRS rerating pipeline (RealtimeReratingGPRS).

```
DataPool
{
     RealtimePipeline
     {
          ModuleName = NET_EM
          Module
          {
               ThreadPool
               {
                   Port = 14579
                   Threads = 4
               }
```

```
                ReratingOpcode
                {
                    OpcodeName = PCM_OP_RATE_PIPELINE_EVENT
                    FieldName = PIN_FLD_EVENT.PIN_FLD_POID
                    GSMEvent
                    {
                      FieldValue = /event/delayed/session/telco/gsm
                      PipelineName = RealtimeReratingGSM
                      NumberOfRTPipelines = 2
                    }
                    GPRSEvent
                    {
                      FieldValue = /event/delayed/session/gprs
                      PipelineName = RealtimeReratingGPRS
                      NumberOfRTPipelines = 2
                    }
                }

            }
        }
}
```

## Adding Real-Time Rerating Pipeline Data to the IFW_PIPELINE Table

Pipeline Manager stores information about pipelines in the IFW_PIPELINE table. The pipelines that are preconfigured in the **Pipelines** section of the default registry file (*Pipeline_home***/conf/ wirelessRealtime.reg**) are inserted into the IFW_PIPELINE table during Pipeline Manager installation.

> **Note:**
>
> - If you are *not* using the default registry file, and you have configured new real-time rerating pipelines, you must manually insert the pipelines into the IFW_PIPELINE table by using SQL commands. Otherwise, you will receive an error at system startup.
> - If you *are* using the default registry file and have changed the default pipeline names or you have configured additional pipelines, you must manually insert the new pipelines into the IFW_PIPELINE table.

The following example shows SQL commands to insert RealtimeReratingGSM and RealtimeReratingGPRS pipelines into the IFW_PIPELINE table:

```
% sqlplus pin@databaseAlias
Enter password: password

SQL>INSERT INTO IFW_PIPELINE ( PIPELINE, NAME, EDRC_DESC ) VALUES
( 'RealtimeReratingGSM', 'GSM Realtime Rerating Pipeline', 'ALL_RATE');

SQL>INSERT INTO IFW_PIPELINE ( PIPELINE, NAME, EDRC_DESC ) VALUES
( 'RealtimeReratingGPRS', 'GPRS Realtime Rerating Pipeline', 'ALL_RATE');

SQL>commit;
```

# Configuring the Batch Rating Pipeline and pin_rerate to Synchronize Processing

To enable **pin_rerate** and the batch rating pipeline to synchronize processes, perform the following tasks:

- Ensure that **pin_rerate** can communicate with a batch pipeline by setting the batch rating pipeline business parameter to **enabled**. See "Specifying Whether the Batch Rating Pipeline Is Enabled".
- Configure rerating business events and event notification. See "Configuring Rerating Business Events and Event Notification".
- Configure Pipeline Manager to send and receive events. See "Configuring Pipeline Manager to Handle Business Events from pin_rerate".
- Configure **pin_rerate** to send and receive events. See "Configuring pin_rerate to Receive Acknowledgment Events from Pipeline Manager".
- Configure standard recycling to enable Pipeline Manager to suspend and recycle EDRs. See "Configuring Standard Recycling".

## Configuring Rerating Business Events and Event Notification

The **pin_rerate** utility and the batch rating pipeline synchronize processing data by passing business events by way of the Oracle DM. To generate business events, **pin_rerate** uses event notification.

The business events and notification events used for rerating are defined in these files:

- The notification events used for rerating are specified by the following entries in the event notification list (*BRM_home***/sys/data/config/pin_notify_ifw_sync**):

```
1301    0    /event/notification/rerating/start
1301    0    /event/notification/rerating/end
1301    0    /event/notification/rerating/PrepareToRerate
1301    0    /event/notification/rerating/ReratingCompleted
```

- The business events used for rerating are specified in the payload configuration file (*BRM_home***/sys/data/config/payloadconfig_ifw_sync.xml**).

## Configuring Pipeline Manager to Handle Business Events from pin_rerate

To enable **pin_rerate** and Pipeline Manager to send and receive business events through the Oracle DM, perform these tasks:

- Creating the Acknowledgment Queue
- Configuring Access to Queues in a Multischema System
- Configuring the DAT_Listener Module

### Creating the Acknowledgment Queue

BRM uses AQ database queues to send business events to Pipeline Manager. A default database queue is created for this purpose when you install BRM.

You must create an additional database queue for Pipeline Manager to post acknowledgment events that are dequeued and processed by **pin_rerate**.

> **Note:**
>
> Only one business event queue and one acknowledgment queue is required for each instance of Pipeline Manager in your system. You do not need to create additional acknowledgment queues if one has already been created.

To create an acknowledgment queue, use the **pin_publish_aq_oracle.pl** utility. For example:

```
pin_publish_aq_oracle.pl create -q ACK_QUEUE -t ack_queue_t -l /@database
```

## Configuring Access to Queues in a Multischema System

You can run BRM and Pipeline Manager in a multischema environment. For example, you might have a schema for BRM that uses the **PIN** logon, a schema for the AQ database queue that uses the **PINQ** logon, and a schema for the instance of Pipeline Manager that uses the **INTEGRATE** logon.

In such cases, you must create a global synonym for the Acknowledgment queue. This enables the user of the BRM schema to access the Acknowledgment queue.

1. Using SQL*Plus, log in to your database as the SYSTEM user:

   ```
   % sqlplus system@DatabaseAlias
   Enter password: password
   ```

2. Create a global user synonym:

   ```
   SQL> CREATE PUBLIC SYNONYM ACCT_SYNC for PINQ.ACCT_SYNC
   ```

   where *PINQ* is the login for the schema that includes the synchronization stored procedures and Acknowledgment queue.

   > **Note:**
   >
   > The rerating stored procedures run in the PIN schema (the BRM schema).

3. Type **exit** to exit SQL*Plus.
4. In the **pin_rerate** configuration file (*BRM_home***/app/pin_rerate/pin.conf**), add the global synonym:

   ```
   - pin_rerate ack_queue PINQ.ACK_QUEUE
   ```

## Configuring the DAT_Listener Module

Pipeline Manager responds to business events sent by **pin_rerate** by posting acknowledgment events. You must configure the DAT_Listener module to post acknowledgment events to the Acknowledgment queue. Set the **AckQueueName** entry to specify the name of the Acknowledgment queue.

Sample DAT_Listener registry:

```
Listener
{
```

```
        ModuleName = DAT_Listener
        Module
        {
             InfranetConnection = ifw.DataPool.LoginQueue
             QueueName = QUEPIN117
             AckQueueName = ACK_QUEUE
             MQAckServerName = ACK_QUEUE_SERVER
             LogEvents = TRUE
        }
}
```

# Configuring pin_rerate to Receive Acknowledgment Events from Pipeline Manager

To enable **pin_rerate** to receive acknowledgment events from Pipeline Manager, perform the following tasks:

- Loading the Stored Procedure for Rerating
- Configuring pin_rerate to Dequeue Events from the Acknowledgment Queue

## Loading the Stored Procedure for Rerating

The **pin_rerate** utility uses stored procedures to dequeue acknowledgment events from the database queue and to purge rerate jobs.

> **Note:**
>
> On Oracle systems, you manually load the rerating stored procedures **job_rerate_procedures_pkb.plb** and **job_rerate_procedures_oracle.plb** into your BRM database.

To load the stored procedure:

1. Connect to your database using SQL*Plus:

   ```
   % sqlplus user@databaseAlias
   Enter password: password
   ```

   where *user* and *password* are your user name and password, and *databaseAlias* is the service name for the BRM database.

2. At the SQL*Plus prompt, enter the following commands:

   ```
   SQL> @BRM_home/sys/dm_oracle/data/job_rerate_procedures_pkg_oracle.plb
   SQL> @BRM_home/sys/dm_oracle/data/job_rerate_procedures_oracle.plb
   ```

   where *BRM_home* is the directory in which BRM is installed.

## Configuring pin_rerate to Dequeue Events from the Acknowledgment Queue

Configure the **pin_rerate** utility to dequeue events from the acknowledgment queue that you created in "Creating the Acknowledgment Queue".

Add the following entry in the **pin_rerate** configuration file (*BRM_home***/app/pin_rerate/pin.conf**):

```
- pin_rerate ack_queue ACK_QUEUE
```

where ACK_QUEUE is the name of the acknowledgment queue.

## Configuring Standard Recycling

BRM rerating uses standard recycling for two purposes:

- To load EDRs suspended during rerating into the BRM database.
- To retrieve suspended EDRs from the BRM database and recycle them when rerating is complete.

# Configuring Rerating When You Do Not Use a Batch Rating Pipeline

If you do not use Pipeline Manager for batch rating, perform the following tasks to configure **pin_rerate** to rerate only real-time-rated events:

- Ensure that **pin_rerate** does not attempt to communicate with a batch rating pipeline by setting the batch rating pipeline business parameter to **disabled**. See "Specifying Whether the Batch Rating Pipeline Is Enabled".
- (Optional) Configure the rerating event cache size. See "Setting the Rerating Event Cache Size (Fetch Size)".
- Configuring the Number of Accounts Per Job and Number of Jobs per Transaction (Optional) Configure parameters for creating rerate jobs. See "Configuring the Number of Accounts Per Job and Number of Jobs per Transaction".

# Specifying Whether the Batch Rating Pipeline Is Enabled

When the batch pipeline is enabled, **pin_rerate** generates notification events to synchronize processing with the batch pipeline by way of the Oracle DM. When the batch pipeline is disabled, **pin_rerate** does not generate notification events because synchronization is not needed.

**pin_rerate** checks a field in the **rerate** instance of the **/config/business_params** object, to determine if the batch rating pipeline is enabled or disabled:

- To rerate both real-time-rated and pipeline-rated events concurrently, the batch rating pipeline entry must be set to **enabled**.
- To rerate only real-time-rated events, the batch rating pipeline entry must be set to **disabled**.

The default is **enabled**.

You modify the **/config/business_params** object by using the **pin_bus_params** utility.

To specify whether the batch pipeline is enabled or disabled:

1. Use this command to create an editable XML file from the **rerate** instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsRerate bus_params_rerate.xml
   ```

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

2. Search the XML file for following line:

   ```
   <BatchRatingPipeline>enabled</BatchRatingPipeline>
   ```

3. Set the entry as needed:

   - **enabled**: To rerate both real-time-rated and pipeline-rated events concurrently.
   - **disabled:** To rerate only real-time-rated events when you do not use Pipeline Manager for batch rating.

   > **Note:**
   >
   > BRM uses the XML in this file to overwrite the existing **rerate** 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 billing configuration.

4. Save the file and change the file name from **bus_params_rerate.xml.out** to **bus_params_rerate.xml**.
5. Use the following command to load the change into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_rerate.xml
   ```

   You should run this command from the *BRM_home***/sys/data/config** directory, which includes support files used by the utility.
6. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.
7. Stop and restart the Connection Manager (CM).

# Setting the Rerating Event Cache Size (Fetch Size)

By default, BRM rerating caches 10,000 events in system memory for processing. Depending on your system memory, you can set the **event_fetch_size** in the Connection Manager's configuration file to specify the number of events retrieved from the database and cached in the system memory for processing.

To set the event cache size:

1. Go to *BRM_home***/sys/data/config**.
2. Use the following command to create an editable XML file from the subscription instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsSubscription bus_params_subscription.xml
   ```

   This command creates an XML file named **bus_params_subscription.xml.out** in your current directory. If you do not want this file in your current directory, specify the path as part of the file name.
3. In **bus_params_subscription.xml.out**, set **EventFetchSize** to the required cache size:

   ```
   <EventFetchSize>value</EventFetchSize>
   ```

   The default value is **10000**.

> **Caution:**
>
> BRM uses the XML in this file to overwrite the existing 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 configuration.

4. Save and exit the file.
5. Rename the **bus_params_subscription.xml.out** file to **bus_params_subcription.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_subcription.xml
   ```

   You should run this command from the *BRM_home***/sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
7. 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 the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.
8. Stop and restart the CM.

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

# Configuring the Number of Accounts Per Job and Number of Jobs per Transaction

By default, BRM assigns 10 accounts to each rerate job and creates 2 rerate jobs per transaction. For example, if the total number of accounts to rerate is 10,000, BRM creates 1000 rerate jobs. It creates the rerate jobs in 500 separate transactions: 2 jobs in each transaction.

To change the default number of accounts per job and the number of rerate jobs created per transaction, perform the following tasks:

1. Open the **pin_rerate** configuration file (*BRM_home***/apps/pin_rerate/pin.conf**).
2. Set the number of accounts assigned to each rerate job by adding the following line:

   ```
   - pin_rerate per_job accounts_per_job
   ```

   where *accounts_per_job* is the number of accounts to assign to each job.
3. Set the number of jobs created per transaction by adding the following line:

   ```
   - pin_rerate per_transaction jobs_per_transaction
   ```

   where *jobs_per_transaction* is the number of jobs to create in each transaction.
4. Save and close the file.

> **Note:**
>
> Setting the **pin_rerate per_job** entry to a small number, for example 1, will result in many rerate jobs being created. Too many rerate jobs can affect your system's performance due to the rerate steps performed for each rerate job. Processing multiple accounts in one rerate job reduces the total number of rerate steps performed compared to processing those same accounts in multiple rerate jobs.

# Configuring Rerating to Reset First-Usage Validity Periods

Perform this task if your product offerings include charge offers, discount offers, or balance impacts that become valid on first usage (when customers use the charge offers and discount offers or consume the balance element balance for the first time).

When rerated events are associated with charge offers, discount offers, or balance element balances that start on first usage, rerating resets their validity periods, if necessary. For example, if the first event rated, which initiated a charge offer's validity period, was not actually the first event to use the charge offer's service, rerating corrects the order of the events and resets the validity period based on the actual first-usage event.

To configure rerating for first-usage validity, perform the tasks in the following sections:

- Configuring the Real-Time Rerating Pipeline to Set Charge Offer Validity Periods.
- Configuring Event Notification for Rerating Cycle Events Triggered by First Usage.

## Configuring the Real-Time Rerating Pipeline to Set Charge Offer Validity Periods

If your charge offers are configured to start when they are first used, configure the ISC_FirstProductRealtime iScript in the real-time rerating pipeline.

When charge offers start on first usage, the real-time rerating pipeline adds the account's first-usage charge offer and discount offer information to the EDR. ISC_FirstProductRealtime sets the validity period in the BRM database for charge offers that were used to rate the event and that start on first usage. This triggers any purchase and cycle fees.

## Configuring Event Notification for Rerating Cycle Events Triggered by First Usage

To ensure that BRM can rerate cycle events that are triggered because of first usage, you must add cycle-event notification events to your event notification list (the default notification list is *BRM_home***/sys/data/config/pin_notify**). BRM provides a list of default cycle-event notification events in the **pin_notify_rerate** file. Use this file to update your **pin_notify** file.

To configure event notification for rerating cycle events triggered by first usage:

1. Open the **pin_notify** and **pin_notify_rerate** files in *BRM_home***/sys/data/config**.
2. Copy the entries in the **pin_notify_rerate** file to the end of the **pin_notify** file.
3. Save and close the **pin_notify** file and close the **pin_notify_rerate** file.
4. Load the contents of the **pin_notify** file into the **/config/notify** object in the BRM database by running the **load_pin_notify** utility.

# Configuring Rerating for Accounts Associated With Subscription Service Transfer

When a subscription service is transferred from one subscriber account to another, rerating the original account can affect the subscription service balances transferred to the new account. However, by default, the rerating process only selects accounts for rerating based on your specified search criteria. This means that, when an account selected for rerating is associated with a subscription service transfer, the other accounts to which the subscription service was transferred are not selected for rerating.

To rerate all accounts associated with a subscription service transfer, you configure the **LineManagement** business configuration parameter. When this parameter is enabled, the rerating process searches for any subscription service transfers for the account that is rerated, and adds the accounts to which the subscription service was transferred during the rerating time specified to the rerate request.

For example:

1. Subscription service X is originally owned by Account A and transferred to Account B on June 15.
2. On June 20, the service is transferred again from Account B to Account C.
3. On July 10, Account A is selected for rerating and the rerating time specified is June 1. Rerating also selects Accounts B and C because the subscription service transfer to Account B and C occurred after June 1. Accounts A, B, and C are grouped together to form a single rerate request.
4. Account A is rerated from June 1 to July 10, Account B is rerated from June 15 to July 10, and Account C is rerated from June 20 to July 10. Rerating selects the events for accounts A, B, and C, and rerates the events in chronological order, resulting in correct subscription service balance updates.

> **Note:**
>
> If rerating fails for any one of the accounts in the rerate request, rerating fails for all the accounts in the rerate request.

During rerating, each account is locked only for as long as it takes to rerate its events. However, in case of a subscription service transfer, related accounts are locked for the duration of rerating events associated with all of those accounts. In the example above, accounts A, B, and C remain locked until rerating of all three accounts is complete.

By default, the **LineManagement** parameter is disabled and cached at CM startup. To change the default value, you modify the **rerate** instance of the **/config/business_params** object by using the **pin_bus_params** utility.

To enable rerating of accounts associated with a subscription service transfer:

1. Run this command to create an editable XML file from the **rerate** instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsRerate bus_params_rerate.xml
   ```

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

2. Search the XML file for following line:

   ```
   <LineManagement>0</LineManagement>
   ```

3. Change **0** to **1**.

   > **Note:**
   >
   > BRM uses the XML in this file to overwrite the existing **rerate** 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 BRM's billing configuration.

4. Save the file and change the file name from **bus_params_rerate.xml.out** to **bus_params_rerate.xml**.
5. Go to the *BRM_home*/**sys/data/config** directory.
6. Run the command:

   **pin_bus_params** *PathToWorkingDirectory*/**bus_params_rerate.xml**

   where *PathToWorkingDirectory* is the directory in which **bus_params_rerate.xml** resides.
7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.
8. Stop and restart the Connection Manager (CM).

# About Automatic Rerating of Backdated Events

> **Note:**
>
> Automatic rerating *does not* mean that rerating occurs immediately when an event is backdated. Automatic rerating means that rerate jobs (**/job/rerate** and **/job_rerate/rerate** objects) are automatically created.

When backdated events occur, BRM uses event notification to automatically create rerate jobs to rerate the backdated events. The accounts in the rerate jobs are rerated the next time you run **pin_rerate** to process rerate jobs; you do not need to specify the accounts and events in the **pin_rerate** command line.

BRM supports automatic rerating of backdated events in the following cases:

- When purchase or cancellation of a charge offer, discount offer, or bundle is backdated.
- When adjustment to a noncurrency balance element is backdated.
- When an extended rating attribute (ERA) modification is backdated.

Table 7-1 shows the entries in the CM configuration file used to create rerate jobs for backdated events.

**Table 7-1 Configuration File Entries for Rerate Jobs**

| Entry | Description |
|---|---|
| **backdate_trigger_auto_rerate** | Specifies whether automatic rerating is enabled. |
| **backdate_window** | Specifies the minimum time difference needed between the current time and the backdated event time for triggering automatic rerating. |
| **num_billing_cycles** | Specifies the maximum number of billing cycles allowed between the current time and the backdated event time of a backdated operation. |

> **Note:**
>
> All of these conditions must be met to trigger automatic rerating. Otherwise, the backdated event is not rerated.

By default, BRM creates rerate jobs for backdated events when the following conditions are met:

- Automatic rerating is enabled.
- The backdated event time is at least one hour earlier than the current time.
- The backdated event date is not older than one billing cycle.

You can change the default backdated time and date thresholds in the CM configuration file. See "Configuring Automatic Rerating of Backdated Events".

You can backdate beyond the number of billing cycles specified in the **num_billing_cycles** entry without requesting to automatically rerate. For more information, see "Backdating beyond Configured Billing Cycles without Automatic Rerating Request".

## About Backdated Bundle, Charge Offer, and Discount Offer Purchase

Automatic rerating of backdated bundle, charge offer, and discount offer purchases is triggered when the difference between the *current time* and the backdated event *purchase, cycle,* or the *usage start time* is greater than the time specified by the **backdate_window** parameter.

BRM then validates that the backdated purchase occurred within the number of billing cycles specified by the **num_billing_cycles** parameter. If the validation is successful, the account is automatically rerated the next time you run **pin_rerate**.

For example, a subscriber's billing day is on the first day of the month. The subscriber purchases a charge offer on July 19 at 5:00 a.m. The CSR backdates the purchase start date to July 15. If the **backdate_window** is set to **1** hour and **num_billing_cycles** is set to **1** cycle, BRM validates that the purchase start time is on or before 4:00 a.m., July 19, and that the purchase start date is not earlier than June 1 (one billing cycle). Because both conditions are met and automatic rerating is enabled, when **pin_rerate** is run with **-rerate** parameter on July 20, BRM automatically rerates all the account's events that occurred after midnight of July 14 to July 20.

# About Backdated Bundle, Charge Offer, and Discount Offer Cancellation

Automatic rerating of backdated bundle, charge offer, and discount offer cancellation is triggered when the difference between the *current time* and the *purchase*, *cycle*, or the *usage end time* is greater than the time specified by the **backdate_window** parameter.

BRM then validates that the backdated cancel occurred within the number of billing cycles specified by the **num_billing_cycles** parameter. If the validation is successful, the account is automatically rerated the next time you run **pin_rerate**.

The following example demonstrates how fees are prorated and charges are reversed when a backdate charge offer cancellation occurs.

In this example, the backdate parameters are as follows:

- **backdate_window** is 2 hours.
- **num_billing_cycles** is 2 cycles.

The subscriber's charge offer includes the following:

- IP charge offer purchase fee: $10
- IP charge offer monthly cycle forward fee: $20 (with proration enabled)
- IP charge offer cancellation fee: $50
- IP usage fee: $1 per minute
- Email charge offer monthly cycle forward fee: $8

1. On January 1, the subscriber purchases the charge offer.

   Account balance = $10 IP charge offer purchase fee + $20 IP monthly cycle forward fee + $8 email cycle forward fee = $38

2. On January 20, the subscriber generates a usage of $10 for the IP service.

   Account balance = $38 + $10 usage = $48

3. On February 1, when billing is run:

   Account balance = $48 + $20 IP monthly cycle forward fee (for February) + $8 email monthly cycle forward fee (for February) = $76

4. On February 10 at 5:00 p.m., the subscriber cancels the IP charge offer. The CSR backdates the charge offer cancellation to January 10. BRM automatically creates a rerate job and the backdate cancel event is rerated by running **pin_rerate**. After backdate cancellation of the IP Charge Offer:

   - A $50 IP charge offer cancellation fee is applied.
   - The IP charge offer monthly cycle forward fee for January is prorated and charged for 10 days only.
   - The $10 IP service usage fee is reversed.
   - The IP charge offer monthly cycle forward fee for February is reversed.

   Account balance = $10 IP purchase fee + $20 IP monthly cycle forward fee (for January) - $14.19 prorated IP monthly cycle forward fee refund + $8 email monthly cycle forward fee (for January) + $50 IP charge offer cancellation fee + $8 email cycle forward fee (for February) = $81.81

5. On March 1, when billing is run:

Account balance = $81.81 + $8 email cycle forward fee (for March) = $89.81

BRM validates that the purchase end time is on or before 3:00 p.m., February 10, and that the purchase end date is not earlier than December 1 (two billing cycles). When both conditions are met and automatic rerating is enabled, when **pin_rerate** is run, BRM automatically rerates all the account's events that occurred after midnight January 10 to February 10.

## About Backdated Adjustment of Noncurrency Balance Elements

When you backdate an adjustment of a noncurrency balance element and automatic rerating is enabled, BRM automatically rerates all the associated events the next time you run **pin_rerate** with the **-rerate** parameter.

> **Note:**
>
> - The adjustment event is rerated only if the adjustment is for a noncurrency balance element.
> - The adjustment must be backdated so that it can be used to rerate the events.

In the following example, the billing date is the first day of the month:

On February 15, the free minutes balance element is adjusted from 100 to 500 and the adjustment is backdated to be effective January 15. The **backdate_window** is 24 hours and **num_billing_cycles** is 1 cycle. When **pin_rerate** is run on February 20 with the **-rerate** parameter, BRM automatically rerates all the relevant events from January 15 to February 20 and applies the balance changes to the current billing cycle. It creates adjustment events for the events that occurred from January 15 to February 1 and shadow events for the events that occurred from February 1 to February 20. Billing events that previously occurred on February 1, such as billing-time discounts, rollovers, and folds, are recalculated based on the new adjustment and are reapplied.

## About Backdated ERA Modifications

BRM automatically rerates backdated ERA modifications when the validity start time or end time is backdated and automatic rerating is enabled.

For example, a subscriber changes their service-level agreement from Silver to Gold on July 12. The CSR backdates the change to July 1 to let the subscriber apply the Gold-level benefits to all usage for July. The next time you run **pin_rerate**, BRM rerates all the relevant events for the account that occurred between July 1 and the current time.

## Configuring Automatic Rerating of Backdated Events

To configure automatic rerating of backdated events, you perform the following tasks:

- Setting Thresholds That Trigger Automatic Rerating
- Configuring Event Notification for Rerating Backdated Events

### Setting Thresholds That Trigger Automatic Rerating

BRM automatically rerates certain backdated events based on the default CM configuration settings. See "About Automatic Rerating of Backdated Events".

To change the default settings:

1. Go to *BRM_home*/**sys/data/config**.
2. Use the following command to create an editable XML file from the subscription instance of the **/config/business_params** object:

   ```
   pin_bus_params -r BusParamsSubscription bus_params_subscription.xml
   ```

   This command creates an XML file named **bus_params_subscription.xml.out** in your current directory. If you do not want this file in your current directory, specify the path as part of the file name.
3. In **bus_params_subscription.xml.out**, set **BackdateTriggerAutoRerate** as required:

   ```
   <BackdateTriggerAutoRerate>enabled</BackdateTriggerAutoRerate>
   ```

   The default value is **disabled**.
4. In **bus_params_subscription.xml.out**, set **BackdateWindow** to the number of seconds required:

   ```
   <BackdateWindow>value</BackdateWindow>
   ```

   The default value is **3600**.
5. In **bus_params_subscription.xml.out**, set **NumBillingCycles** as required:

   ```
   <NumBillingCycles>value</NumBillingCycles>
   ```

   The default value is **1**.
6. **Caution:**

   BRM uses the XML in this file to overwrite the existing 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 configuration.
7. Save and exit the file.
8. Rename the **bus_params_subscription.xml.out** file to **bus_params_subscription.xml**.
9. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_subcription.xml
   ```

   You should run this command from the *BRM_home*/**sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
10. 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 the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.
11. Stop and restart the CM.

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

## Configuring Event Notification for Rerating Backdated Events

When a backdated event occurs, BRM rerating uses event notification to trigger automatic rerating of the event.

Although any subclass of the **/event** class can be used to trigger event notification, BRM rerating generates the nonpersistent **/event/notification/auto_rerate** event specifically to use for event notification.

By default, when this event occurs, BRM creates a rerate job.

Before you can use BRM rerating, you must configure the event notification feature as follows:

1. If your system has multiple configuration files for event notification, merge them.
2. Ensure that the merged file includes the following information from the *BRM_home***/sys/data/config/pin_notify** file:

```
# Rerating related event notification
3787    0    /event/notification/auto_rerate
```

3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list.
4. (Optional) If necessary to accommodate your business needs, create custom code for event notification to trigger.
5. Load your final event notification list into the BRM database.

# Backdating beyond Configured Billing Cycles without Automatic Rerating Request

You can backdate an event beyond the configured number of billing cycles without requesting to automatically rerate such an event.

To do so:

1. Ensure that automatic rerating is enabled for backdated events. If necessary, enable it by setting the **backdate_trigger_auto_rerate** entry in CM configuration file to **1**.
2. Enable the **AllowBackdateNoRerate** business parameter in the **/config/business_params** object by using the **pin_bus_params** utility.
3. After you enable the **AllowBackdateNoRerate** business parameter, you must manually rerate any events backdated *beyond* the number of billing cycles specified in the **num_billing_cycles** entry.

## Enabling the AllowBackdateNoRerate Business Parameter

To set the **AllowBackdateNoRerate** business parameter to **enabled**:

1. Go to *BRM_home***/sys/data/config** directory.
2. Create an editable XML file of the subscription instance from the **/config/business_params** object by using the following command:

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

BRM places the XML file named **bus_params_subscription.xml.out** in your working directory. To place this file in a different directory, specify the full path as part of the file name.

3. Locate the **AllowBackdateNoRerate** entry in the **bus_params_subscription.xml.out** file.
4. Set the value of **AllowBackdateNoRerate** to **enabled**, if necessary.

   ```
   <AllowBackdateNoRerate>enabled</AllowBackdateNoRerate>
   ```

   > **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 subscription configuration.

5. Save this updated file as **bus_params_subscription.xml**.
6. Load the modified XML file containing the business parameters for billing into the appropriate **/config/business_params** object in the BRM database.

   ```
   pin_bus_params bus_params_subscription.xml
   ```

   You should run this command from the *BRM_home***/sys/data/config** directory, which includes support files used by the utility.
7. Read the object with the **testnap** utility or Object Browser to verify that all fields are correct. BRM stores one of the following values for **AllowBackdateNoRerate**:
   - **0** to indicate **disabled**.
   - **1** to indicate **enabled**.
8. Stop and restart Connection Manager.

# About Automatic Rerating of Out-of-Order Events

Events are processed by Pipeline Manager in the order that call details records (CDRs) are received. If the CDRs are sent out of order to Pipeline Manager, the events are processed out of order as well. Usually, this is not a problem; however, correct rating sometimes depends on rating events in chronological order (for example, when usage counters are used).

You can configure BRM to detect events that must be rated in chronological order and rerate them. To use out-of-order rerating, you define the criteria for when an out-of-order event must be rerated. When an event is rated, the FCT_EventOrder module uses the criteria and the event timestamps to determine if the event needs to be rerated.

## About Detecting Out-of-Order Events

To enable out-of-order rerating, you configure criteria BRM uses to detect when events qualify for out-of-order rerating. Events must be rated in chronological order only when balance elements from the same balance group are affected, so the balance group is assumed as part of the detection criteria. If an account owns more than one service instance, each using a different balance group, out-of-order detection is applied to only the balance group associated with the event (the criteria name and balance group combination that is stored in the **/profile/event_ordering** object).

Pipeline Manager loads your detection criteria configuration at startup as follows:

- The DAT_PortalConfig module retrieves data from the **/config/event_order_criteria** object and loads it into its memory. This data specifies the criteria for determining if an event qualifies for out-of-order detection. The criteria is based on:
  - Service types
  - Event types
  - Charge offers
  - Discounts
  - A combination of service types, event types, charge offers, and discounts

  For more information, see "About Out-of-Order Rerating Criteria".

  > **Note:**
  >
  > This feature supports branding. You can have a different configuration for each brand.

- The DAT_AccountBatch module retrieves data from the **/profile/event_ordering** profile object and loads it into its memory. This data includes the timestamp for the last time an event was processed for a particular billing cycle with the criteria specified in the **/config/event_order_criteria** object and the balance group determined when an EDR is found to be out of order.

  > **Note:**
  >
  > If an out-of-order event comes in from a previous billing cycle after billing has run for that cycle (for example, when delayed billing is configured), the event is handled by default as if it is part of the current billing cycle.

## How BRM Rerates Out-of-Order Events

The overall process for rerating out-of-order events is as follows:

1. Pipeline Manager loads your detection criteria configuration at startup. See "About Detecting Out-of-Order Events".
2. An event is rated in the pipeline and is rated by the rating and discounting modules.
3. The FCT_EventOrder module gets the out-of-order detection criteria from the DAT_PortalConfig module to determine if the event needs to be rerated.

   In addition, FCT_EventOrder gets data from the DAT_AccountBatch module that specifies the latest event processed time for each appropriate criterion and balance group combination for an active billing cycle.
4. FCT_EventOrder determines whether the event needs to be rerated:
   - If the event is out of order, the module proceeds to the next step.
   - If the event is not out of order, FCT_EventOrder triggers an update to the last event processed time in the DAT_AccountBatch memory. A record (record type = 850) is added to the pipeline output to update the **/profile/event_ordering** object with the new event's start time.

5. The event data record (EDR) is kept in the FCT_EventOrder module's shared memory until after the transaction commits, when it is then sent to a rerate-request file. When Batch Controller detects this file, it starts the OODHandler batch handler to process it. The rerate-request file contains multiple accounts for which out-of-order events were detected.

> **Note:**
>
> - You can configure FCT_EventOrder to write the out-of-order EDR data to separate rerate-request files after the transaction commits; batching the requests in this way helps performance in Pipeline Manager. See "Batching Out-of-Order Rerate Jobs".
> - You can configure FCT_EventOrder to write a specific number of accounts to each rerate-request file. See "Configuring the Number of Accounts in an Out-of-Order Rerate Job".

Rerate-request file names use the following format:

*outputPrefix_pipelineName_transactionID_sequenceNumber*.**xml**

where:

*outputPrefix* is the prefix specified by the **OutputPrefix** entry in the FCT_EventOrder module registry. The default is **ood**.

*pipelineName* is the name of the pipeline; in the following example, the name is ALL_RATE.

*transactionID* is the transaction ID.

*sequenceNumber* is the sequence number of the job.

For example:

```
ood_ALL_RATE_14_0.xml
```

6. The OODHandler batch handler processes the out-of-order rerate-request file, moves it to the *BRM_home*/**apps/pin_ood_handler/process** directory, and calls the **pin_load_rerate_jobs** utility.
7. The **pin_load_rerate_jobs** utility creates an input flist with the following rerate output file data:
   - Account
   - Balance Group
   - CriteriaName
   - Rerate Start Time
   - Service

   It then calls the PCM_OP_ACT_USAGE opcode in calc-only mode, which generates a notification event (**/event/notification/activity/out_of_order**) that triggers the PCM_OP_ACT_HANDLE_OOD_EVENT opcode. PCM_OP_ACT_USAGE passes the data to PCM_OP_ACT_HANDLE_OOD_EVENT.
8. PCM_OP_ACT_HANDLE_OOD_EVENT calls the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode.

9. PCM_OP_RERATE_INSERT_RERATE_REQUEST checks for duplicate rerate jobs in the rerate-request file and calls other opcodes to create a rerate job out of the file.
10. The rated event is loaded into the BRM database by Rated Event (RE) Loader. If the event qualifies for out-of-order detection (and the event is in order), RE Loader updates the account's **/profile/event_ordering** object.

> **Note:**
>
> The **/profile/event_ordering** object stores data for the current and next billing cycles and for closed billing cycles. To clean up data for closed billing cycles, see "Purging Event Ordering Profile Data for Closed Billing Cycles".

## About Out-of-Order Rerating Criteria

When the FCT_EventOrder module checks for out-of-order events, it uses the following criteria:

- The timestamp of the event. If the event is later than the latest event that is under consideration, the event is not out of order.
- The out-of-order detection criteria in which the event is defined in the **/config/event_order_criteria** object. If the event is not defined in any criteria in the **/config/event_order_criteria** object, it is assumed the event does not need to be rated in chronological order.
- The last event processed time for each balance group and the name of the trigger-dependant rerating criteria for that event for a billing cycle in the account's **/profile/event_ordering** object. If the event has a later timestamp than the latest event processed time for an event that uses the same balance group and criteria name, the event is not out of order.

Define the criteria for the events you must rate in chronological order in the **/config/event_order_criteria** object. The data stored in the **/config/event_order_criteria** object consists of two parts:

- The name of the criterion.
- One or more parameters, such as a service or event type or a charge offer name or discount name.

Your configuration can use as many parameters as you require. Table 7-2 provides examples of out-of-order criteria and what they mean for out-of-order detection.

**Table 7-2 Criteria for Out-of-Order Detection**

| Criteria | Out-of-Order Detection |
|---|---|
| Criteria name: GSM_TEL_SERVICE<br>Parameter: **/service/gsm/telephony** | All events for the GSM telephony service should be considered for out-of-order rerating. |
| Criteria name: GSM_EVENTS<br>Parameter: **/event/delayed/session/telco/gsm/telephony**<br>Parameter: **/event/delayed/session/telco/gsm/sms**<br>Parameter: **/event/delayed/session/telco/gsm/fax** | All GSM events should be considered for out-of-order rerating. |

**Table 7-2 (Cont.) Criteria for Out-of-Order Detection**

| Criteria | Out-of-Order Detection |
|---|---|
| Criteria name: GSM_Product<br>Parameter: GSM Voice Product | All events for a specific charge offer should be considered for out-of-order rerating.<br>All EDRs that are rated by the GSM Voice Charge Offer will be considered for out-of-order rerating. |

For detailed information on defining your criteria, see "Defining Out-of-Order Criteria".

# Setting Up Out-of-Order Rerating

To set up out-of-order rerating, perform the tasks in the following sections:

- Defining Out-of-Order Criteria
- Loading Out-of-Order Criteria

> **Note:**
>
> Accounts created *after* you load the **/config/event_order_criteria** object in the BRM database qualify for out-of-order detection.

- Configuring Out-of-Order Detection in a Pipeline.
- Configuring Event Notification for Out-of-Order Rerating
- (Optional) "Specifying a Reason Code for Rerating Out-of-Order Events"
- Configuring Batch Controller for Rerating Out-of-Order Events
- Configuring the OODHandler Batch Handler for Rerating Out-of-Order Events
- (Optional) "Purging Event Ordering Profile Data for Closed Billing Cycles"

## Defining Out-of-Order Criteria

When you define out-of-order criteria, you specify parameters for events, services, discount names, or charge offer names that qualify for out-of-order detection. The parameters you specify are the criteria BRM uses to ensure that events are rated in chronological order when you want them to be. For more information, see "About Out-of-Order Rerating Criteria".

> **Note:**
>
> You should know how your offers are structured to track balance elements, such as balance elements for a specific service or balance elements for a specific type of usage. Tracking balance elements can involve using service-level balance groups and a criterion is set up for events that use the same balance group.

You typically list parameters that use the same balance group under the same criterion name. Services that use different balance groups are listed under separate criterion names. The **/profile/event_ordering** object stores the latest event processed time for each appropriate criteria and balance group combination for each active billing cycle. For more information, see "Defining Criteria under Separate Criteria Names".

You define out-of-order criteria in the **pin_config_ood_criteria.xml** file in *BRM_home***/sys/data/config**. For a sample of this file, see "Sample Out-of-Order Criteria File".

You can define criteria by using one or more parameters.

- The simplest criteria uses a single parameter.

  For example:

```
- <OodCriteriaElement>
<CriteriaName>GSM_TEL_SERVICE</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/gsm/telephony</Parameter>
</ParameterList>
</OodCriteriaElement>
```

- You can use two parameters in the following combinations:
  - Service Type, Event Type
  - Service Type, Product
  - Service Type, Discount
  - Event Type, Product
  - Event Type, Discount

  For example:

```
- <OodCriteriaElement>
<CriteriaName>GSM_TEL_SERVICE</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/gsm/telephony</Parameter>
<Parameter Type="Event">/event/delayed/session/gsm/telephony</Parameter>
</ParameterList>
</OodCriteriaElement>
```

- You can use multiple parameters to combine service type, event type, charge offer, and discount.

  For example:

```
- <OodCriteriaElement>
<CriteriaName>GSM</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
<Parameter Type="Service">/service/telco/gsm/sms</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/telephony</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/sms</Parameter>
<Parameter Type="Product">GSM Voice Product</Parameter>
<Parameter Type="Product">GSM Voice Discount</Parameter>
</ParameterList>
</OodCriteriaElement>
```

If you use a combination of event, service, and charge offer or discount, the EDR parameters must match all of the following:

- One service type (for example, service type 1 or service type 2)
- One event type (for example, event type 1 or event type 2)

- One charge offer or discount offer (for example, charge offer 1 or charge offer 2 or discount offer 1 or discount offer 2)

For example, in the following detection criterion named GSM_Multi:

```
- <OodCriteriaElement>
<CriteriaName>GSM_Multi</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
<Parameter Type="Service">/service/telco/gsm/sms</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/telephony</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/sms</Parameter>
<Parameter Type="Product">GSM Voice Product Telephony</Parameter>
<Parameter Type="Discount">GSM Voice Product SMS</Parameter>
</ParameterList>
</OodCriteriaElement>
```

The EDR must match the following parameters:

- Service type **/service/telco/gsm/telephony**
- Event type **/event/delayed/session/telco/gsm/telephony**
- Charge Offer GSM Voice Charge Offer Telephony

Or the following parameters:

- Service type **/service/telco/gsm/sms**
- Event type **/event/delayed/session/telco/gsm/sms**
- Charge Offer GSM Voice Charge Offer SMS

That means that when an EDR arrives for that particular service rated by that particular charge offer when that particular event occurs, BRM considers it for out-of-order rerating.

## Defining Criteria under Separate Criteria Names

List services that use the same balance group under the same criteria name. BRM does not differentiate between the events of these services when obtaining the latest event processed time in the account's profile object because they consume balance elements in the same way for rating.

The following out-of-order criteria would apply to two GSM services that share the same balance group:

```
- <OodCriteriaElement>
<CriteriaName>GSM</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
<Parameter Type="Service">/service/telco/gsm/sms</Parameter>
</ParameterList>
</OodCriteriaElement>
```

If the latest event processed time is later than an EDR event timestamp, the event needs to be rerated regardless of the service to which it belongs; for example, for the preceding GSM criteria, if the latest event processed time is 2 p.m. for an SMS event and the EDR is a telephony event with a 1 p.m. timestamp, the event is out of order and needs to be rerated.

If an account can own more than one instance of a service and each instance uses a different balance group, out-of-order rerating is applied to the balance group of the service instance to which the event belongs. Accounts can have multiple balance groups if you offer packages in which service-level balance groups are used. When events for these services must be rated in

chronological order, you must evaluate the out-of-order criteria for these services separately from each other.

You list services that use different balance groups under separate criteria names. The following out-of-order criteria would apply to two GSM services that use different balance groups:

```
- <OodCriteriaElement>
<CriteriaName>GSM_TEL</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
</ParameterList>
</OodCriteriaElement>
- <OodCriteriaElement>
<CriteriaName>GSM_SMS</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/sms</Parameter>
</ParameterList>
</OodCriteriaElement>
```

Because the latest event processed times for all services used by an account are tracked in one profile object, BRM uses the unique criteria name to differentiate between the service-specific latest event processed times. Thus, in the preceding criteria, BRM uses the criteria name GSM_TEL in the account's profile object to track the latest event processed time for the telephony events and uses the criteria name GSM_SMS to track the latest event processed time for the SMS events.

There may be times when you must define parameters under separate criterion names when they share the same balance group but you are tracking different balance elements. For example, if a GSM telephony service and a GSM discount share a balance group and are rated by the same event, you would define a criterion for the discount if you are tracking the last time balance elements were consumed with that particular discount.

## Avoiding Overlapping Criteria

If you configure criteria already contained in another criteria, the **load_pin_config_ood_criteria** utility reports an error, and nothing is loaded into the database. In the following example, the GSM_TEL_1 criteria is a superset of the GSM_TEL_2 criteria, so GSM_TEL_2 is considered an overlapping criteria:

```
- <OodCriteriaElement>
<CriteriaName>GSM_TEL_1</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
</ParameterList>
</OodCriteriaElement>
- <OodCriteriaElement>
<CriteriaName>GSM_TEL_2</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/telephony</Parameter>
</ParameterList>
</OodCriteriaElement>
```

In the following example, the GSM_Dual_A criteria is a superset of the GSM_Multi_B criteria, so GSM_Multi_B is an overlapping criteria:

```
- <OodCriteriaElement>
<CriteriaName>GSM_Dual_A</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
<Parameter Type="Service">/service/telco/gsm/sms</Parameter>
```

```
<Parameter Type="Product">GSM Voice Product Telephony</Parameter>
<Parameter Type="Discount">GSM Voice Discount SMS</Parameter>
</ParameterList>
</OodCriteriaElement>
- <OodCriteriaElement>
<CriteriaName>GSM_Multi_B</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
<Parameter Type="Service">/service/telco/gsm/sms</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/telephony</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/sms</Parameter>
<Parameter Type="Product">GSM Voice Product Telephony</Parameter>
<Parameter Type="Discount">GSM Voice Discount SMS</Parameter>
</ParameterList>
</OodCriteriaElement>
```

You would choose one of the preceding criterion based on your goals. You would never use both of them. For example:

- You would configure the GSM_Dual_A criterion if you wanted BRM to consider EDRs for out-of-order rerating when they matched the following:
  - Any event type for the GSM telephony service that is also rated by the GSM Voice Charge Offer Telephony charge offer.

  or

  - Any event type for the GSM SMS service that is also rated by the GSM Voice Discount SMS discount.
- You would configure the GSM_Multi_B criterion if you wanted BRM to consider EDRs for out-of-order rerating when they matched the following:
  - The specific event type **/event/delayed/session/telco/gsm/telephony** for the GSM telephony service that is also rated by the GSM Voice Charge Offer Telephony charge offer.

  or

  - The specific event type **/event/delayed/session/telco/gsm/sms** for the GSM SMS service that is also rated by the GSM Voice Discount SMS discount.

Because GSM_Dual_A is configured to consider all event types including the specific event types configured in GSM_Multi_B, the out-of-order detection configured in GSM_Multi_B is already handled by the GSM_Dual_A criterion. If you only want those specific event types to be checked for out-of-order detection, you would use GSM_Multi_B rather than GSM_Dual_A.

## Sample Out-of-Order Criteria File

The following is a sample configuration of the **pin_config_ood_criteria.xml** file.

```
<?xml version="1.0" encoding="UTF-8" ?>
- <OodCriteriaConfiguration xmlns="http://www.portal.com/schemas/BusinessConfig"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://
www.portal.com/schemas/BusinessConfig pin_config_ood_criteria.xsd">

- <!--  This file is a sample file. -->
- <!--  This file needs to be modified based on the out-of-order configuration required
-->
- <!--  for an installation. THIS FILE SHOULD NOT BE LOADED WITH THE DATA SUPPLIED HERE.
-->

- <OodCriteriaElement>
<CriteriaName>TEL_SERVICE</CriteriaName>
```

```
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/telephony</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/telephony</Parameter>
</ParameterList>
</OodCriteriaElement>

- <OodCriteriaElement>
<CriteriaName>GSM_SERVICE</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm/sms</Parameter>
<Parameter Type="Event">/event/delayed/session/telco/gsm/sms</Parameter>
</ParameterList>
</OodCriteriaElement>

- <OodCriteriaElement>
<CriteriaName>A_MULTI_TEL_SERVICE</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm</Parameter>
<Parameter Type="Product">Standard GSM Telephony</Parameter>
</ParameterList>
</OodCriteriaElement>

- <OodCriteriaElement>
<CriteriaName>A_MULTI_TEL_SERVICE1</CriteriaName>
- <ParameterList>
<Parameter Type="Service">/service/telco/gsm</Parameter>
<Parameter Type="Discount">Standard GSM Discount</Parameter>
</ParameterList>
</OodCriteriaElement>

- <OodCriteriaElement>
<CriteriaName>TEL_SERVICE_EVENT</CriteriaName>
- <ParameterList>
<Parameter Type="Event">/event/delayed/session/telco/gsm</Parameter>
</ParameterList>
</OodCriteriaElement>

</OodCriteriaConfiguration>
```

## Loading Out-of-Order Criteria

To configure out-of-order criteria, you edit the **pin_config_ood_criteria.xml** file and load the contents of the file into the BRM database by using the **load_pin_config_ood_criteria** utility. The data is stored in the **/config/event_order_criteria** object.

> **Note:**
>
> - To connect to the BRM database, the **load_pin_config_ood_criteria** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility.
> - The **load_pin_config_ood_criteria** utility overwrites existing out-of-order criteria. If you are updating out-of-order criteria, you cannot load new out-of-order criteria only. You must load complete sets of out-of-order criteria each time you run the **load_pin_config_ood_criteria** utility.
> - You can run this utility to configure out-of-order criteria for different brands.

1. Edit the **pin_config_ood_criteria.xml** file in *BRM_home*/**sys/data/config**. For a sample of this file, see "Sample Out-of-Order Criteria File".
2. Save and close the file.
3. Use the following command to run the **load_pin_config_ood_criteria** utility:

   ```
   load_pin_config_ood_criteria pin_config_ood_criteria.xml
   ```

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

   ```
   load_pin_config_ood_criteria BRM_home/sys/data/config/pin_config_ood_criteria.xml
   ```

   > **Note:**
   >
   > If you copy the **pin_config_ood_criteria.xml** file to the directory from which you run the **load_pin_config_ood_criteria**utility, you do not have to specify the path or file name. By default, the file is named **pin_config_ood_criteria.xml**. You can change this name.

4. If Pipeline Manager is running, send a CBPReload semaphore to reload data. The **Reload** semaphore is used by the DAT_PortalConfig data module. If Pipeline Manager is not running, it will load the data the next time it is started.

To verify that the out-of-order criteria were loaded, you can display the **/config/event_order_criteria** object by using Object Browser, or use the **robj** command with the **testnap** utility.

## Configuring Out-of-Order Detection in a Pipeline

To configure out-of-order detection in a pipeline, you do the following:

1. Configure the DAT_PortalConfig module.
2. Configure the FCT_EventOrder module.

   > **Note:**
   >
   > FCT_EventOrder should be located in the pipeline *after* the rating (FCT_MainRating) and discounting (FCT_Discount) modules and *before* the rejection (FCT_Reject) module.

   When configuring FCT_EventOrder, specify the following:

   - The amount of data to include in each rerate request file. See "Batching Out-of-Order Rerate Jobs".
   - The number of accounts assigned to each rerate job. See "Configuring the Number of Accounts in an Out-of-Order Rerate Job".

### Batching Out-of-Order Rerate Jobs

You can configure the FCT_EventOrder module to batch the out-of-order EDR data in its transactional memory and write it to separate rerate-request files after the transaction commits. Batching rerate jobs in this way improves performance during rerating. To control how many

records FCT_EventOrder batches into separate rerate request files, you specify an amount of time in minutes using the **RerateDelayTolerance** registry entry.

When the transaction commits, FCT_EventOrder sorts the rerate-request data in its transactional memory based on the EDR start time. The rerate start time of the first record in the rerate-request file is the rerate start time for the rerate job. As FCT_EventOrder writes the events into the rerate-request file, it compares the start time of the first record with the start time of each event being added. If the time difference exceeds the value of the **RerateDelayTolerance** entry, that event becomes the last record in the rerate-request file, and a new rerate-request file is created.

Example 1

If the **RerateDelayTolerance** entry is set to 180 minutes, and the start time of the first event in the rerate-request file is 3:15 p.m, events are added until the start time of an event is greater than 6:15 p.m.

Example 2

If the **RerateDelayTolerance** entry is set to 30 minutes, and the start time of the first event in the rerate-request file is 11:00 a.m. and four subsequent events have start times of 11:10 a.m., 12:05 p.m., 12:15 p.m., and 12:33 p.m, FCT_EventOrder creates two rerate-request files, resulting in two rerate jobs (**/job_batch/rerate** objects) as follows:

- Rerate job 1 contains accounts associated with EDRs that have these start times:

  11:00 a.m.

  11:10 a.m.

- Rerate job 2 contains accounts associated with EDRs that have these start times:

  12:05 p.m.

  12:15 p.m.

  12:33 p.m.

In the preceding example, if for one account the first EDR arrives at 11:10 a.m. and the last EDR arrives at 11:45 a.m., the account is included in Rerate job 1.

## Configuring the Number of Accounts in an Out-of-Order Rerate Job

To improve batch rerating throughput, you can specify the number of accounts FCT_EventOrder assigns to each rerate job by using the **NumberOfAccountLimit** registry entry. FCT_EventOrder writes out the number of accounts specified by this entry to the rerate-request file, which, in turn, becomes the rerate job after detection of duplicate rerate requests is complete.

The default for the **NumberOfAccountLimit** registry entry is 1000.

> **Note:**
>
> The **NumberOfAccountLimit** registry entry of FCT_EventOrder is similar to the **per_job** configuration entry of the **pin_rerate** utility. When you configure these entries, BRM recommends you use the same value.

For more information, see "Configuring the Number of Accounts Per Job and Number of Jobs per Transaction".

## Configuring Event Notification for Out-of-Order Rerating

To create rerate jobs automatically when certain events are not in chronological order, you must configure event notification for out-of-order rerating. BRM uses the **/event/notification/activity/out_of_order** event to trigger automatic rerating when an out-of-order event occurs.

To configure event notification for out-of-order rerating, do the following:

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

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

   Depending on which BRM features you use, your system may contain one or more configuration files for event notification; for example, a wireless system may use **pin_notify_ifw_sync** (supports account synchronization and standard recycling) or **pin_notify_telco** (supports GSM Manager). If your system contains more than one of these files, you must merge their contents into a single file.

2. Add the following entry to the merged file:

   ```
   # Event notification for out-of-order rerating
   177    0    /event/notification/activity/out_of_order
   ```

   This configures the **/event/notification/activity/out_of_order** event to trigger rerating and to call the PCM_OP_ACT_HANDLE_OOD_EVENT opcode (opcode ID number 177) to select which events to rerate.

   PCM_OP_ACT_HANDLE_OOD_EVENT calls the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode to insert a rerate job for the out-of-order events.

   > **Note:**
   >
   > You can configure this notification event to call a custom opcode. The custom opcode can analyze the event, determine if rerating is required, and then call PCM_OP_RERATE_INSERT_RERATE_REQUEST to create the rerate job with a rerate reason. For more information, see "Setting Up Trigger-Dependent Rerating".

3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list.
4. (Optional) If necessary to accommodate your business needs, create custom code for event notification to trigger.
5. Load your final event notification list into the BRM database by running the **load_pin_notify** utility to load the contents of the file into the **/config/notify** object.

## Specifying a Reason Code for Rerating Out-of-Order Events

Rerate jobs can be assigned a rerate reason code. When you run the **pin_rerate** utility, you can rerate all accounts for a rerate job with a specific reason code. By default, rerate jobs for out-of-order events have the reason code **1**. You can change this to any number you want in the CM **pin.conf** file by using the following entry:

```
fm_act ood_rerate_job_reason_code rerate_reason_code
```

> **Note:**
>
> Rerate reason codes can also be assigned by other BRM automatic rerating features and when you run **pin_rerate**. To rerate accounts according to type of rerate reason code, ensure that your rerate reason codes are unique.

## Configuring Batch Controller for Rerating Out-of-Order Events

The OODHandler batch handler is run automatically by Batch Controller. You must configure Batch Controller and the OODHandler batch handler.

To configure Batch Controller to rerate out-of-order events:

1. Open the Batch Controller **Infranet.properties** file in *BRM_home***/apps/batch_controller**.
2. Add the following entries listed in Table 7-3 for the **OODHandler** batch handler.

**Table 7-3 Entries for OODHandler Batch Handler**

| Entry | Description |
|---|---|
| **batch.random.events** | Specify your event identifier in the event identifier list. For example:<br>`batch.random.events TEL, SMS, FAX, OODHANDLER`<br>where **OODHANDLER** is your event identifier. |
| *event_identifier***.name** | Specify a description for the event identifier. For example:<br>`OODHANDLER.name OODHANDLER usage` |
| *event_identifier***.file.location** | Specify the path to the directory where Pipeline Manager puts the rerate request files. The location of this directory is configured in the **OutputDirectory** entry in the FCT_EventOrder module wireless registry file (*Pipeline_home***/conf/wireless.reg**).<br>**Important:** You must create the directory. It is not created by BRM installation scripts.<br>For example:<br>`OODHANDLER.file.location Pipeline_home/data/out/ood` |
| *event_identifier***.file.pattern** | Specify the rerate job output file name. When Batch Controller detects a file with this name, it starts the batch handler.<br>**Tip:** You can use an asterisk (*) to represent zero or more characters in the file name. No other wildcards are supported.<br>For example:<br>`OODHANDLER.file.pattern *.xml` |

**Table 7-3 (Cont.) Entries for OODHandler Batch Handler**

| Entry | Description |
|---|---|
| *event_identifier*.**handlers** | Specify the batch handler identifier. For example:<br>`OODHANDLER.handlers OodHandler` |
| *handler_identifier*.**name** | Specify a description for the batch handler identifier. For example:<br>`OodHandler.name OODHandler` |
| *handler_identifier*.**max.at.lowload.time**<br>*handler_identifier*.**max.at.highload.time** | Specify the number of batch handler instances that can run concurrently during periods of low load and high load usage. Typical default settings are 6 at low load and 3 at high load. For example:<br>`OodHandler.max.at.lowload.time 6`<br>`OodHandler.max.at.highload.time 3` |
| *handler_identifier*.**start.string** | Specify the command that starts the OODHandler batch handler. For example, the default is:<br>`OodHandler.start.string` *BRM_home*`/apps/pin_ood_handler/OODHandler.pl.`<br>**Important:** Copy the *BRM_home*/**bin/OODHandler** to *BRM_home*/**apps/pin_ood_handler/OODHandler.pl**. |

3. Save and close the file.

## Configuring the OODHandler Batch Handler for Rerating Out-of-Order Events

The OODHandler batch handler retrieves the rerate-request file from the Pipeline Manager output and runs the **pin_load_rerate_jobs** utility to create rerate jobs. After the rerate jobs are created, the batch handler moves the rerate-request file to a different directory.

> **Note:**
>
> If you use the ConfigurableValidityHandler batch handler for loading first-usage validity data, do not use the **OODHandler_config.values** file as instructed below. Instead, you must configure the OODHandler batch handler in the ConfigurableValidityHandler configuration file (*BRM_home*/**apps/pin_rel/ConfigurableValidityHandler_config.values**). ConfigurableValidityHandler runs both the **pin_load_rerate_jobs** utility and the utility for loading validity data.

To configure the OODHandler batch handler:

1. Open the **OodHandler_config.values** configuration file in *BRM_home*/**apps/pin_ood_handler**.
2. Edit the following entries shown in Table 7-4. For information about other entries, see the **OodHandler_config.values** file.

**Table 7-4 Entries to Configure OodHandler Batch Handler**

| Entry | Description |
|---|---|
| **$FILETYPE** | Specify the file name pattern of the rerate-request file. For example:<br>`$FILETYPE = "*.xml.bc";`<br>**Note:** The asterisk (*) represents zero or more characters in the file name. No other wildcards are supported.<br>Batch Controller runs the OODHandler batch handler for each file with a name that matches this pattern.<br>**Important:** The file name pattern must end with the **.bc** extension. Batch Controller automatically appends **.bc** to each file name before it runs a batch handler. |
| **$HANDLER_DIR** | Specify the path to the directory containing the OODHandler batch handler configuration files, log files, and other processing files and directories.<br>The default is *BRM_home***/apps/pin_ood_handler**. |
| **$pinLoadRerateJobsDir** | Specify the path to the directory where the **pin_load_rerate_jobs** utility processes the files (this contains the **pin_rerate_job_info.xsd** file).<br>The default is *BRM_home***/apps/pin_ood_handler/process**. |
| **$pinLOADRERATEJOBS** | Specify the application run by the OODHandler batch handler to process the files. For example:<br>`$pinLOADRERATEJOBS = "pin_load_rerate_jobs";` |
| **$STAGING** | Specify the full path to the OODHandler batch handler rerate-request file location.<br>For example:<br>`$STAGING = "`*`Pipeline_home`*`/data/out/ood";`<br>This is the same directory where Pipeline Manager puts the rerate-request files. When Batch Controller detects the rerate-request files in this location, it calls the OODHandler batch handler.<br>**Note:** The location of this directory is also configured in the **OutputDirectory** entry in the FCT_EventOrder module wireless registry file (*Pipeline_home***/conf/wireless.reg**). You must create the directory. It is not created by BRM installation scripts. |
| **$PROCESSING** | Specify the full path to the directory from which the rerate job files are processed by the OODHandler batch handler.<br>The default is **$pinLoadRerateJobsDir**.<br>The OODHandler batch handler takes the files from the **$STAGING** directory and places them here. |
| **$LOGFILE** | Specify the full path to the OODHandler batch handler log file.<br>For example:<br>`$LOGFILE = "$HANDLER_DIR/OOD_Handler.log";` |

3. Save and close the file.

## Purging Event Ordering Profile Data for Closed Billing Cycles

If out-of-order detection is configured in the system, one instance of the **/profile/event_ordering** object is created for each account during customer creation. The profile exists

for the lifetime of the account, and entries are updated and added to it by RE Loader every time events are processed for services belonging to the account that qualifies for out-of-order detection.

You can use the **purge_profile_event_ordering** script to clean up entries for closed billing cycles from the **/profile/event_ordering** object as follows:

Log on to the machine where **dm_oracle** is installed and run the Shell script:

```
BRM_home/apps/pin_rel/purge_profile_event_ordering
```

The **purge_profile_event_ordering** script is located in *BRM_home*/**apps/pin_rel**.

> **Note:**
>
> Cleaning up data for closed billing cycles is not automatically synchronized with Pipeline Manager. You must restart Pipeline Manager for the in-memory entries to be cleaned up.

# About Trigger-Dependent Rerating

Trigger-dependent rerating enables you to specify when events are automatically rerated. For example, if you rerate usage following a charge offer cancellation, you can set up a charge offer cancellation event to automatically trigger rerating.

To configure trigger-dependent rerating, you set up event notification to call a custom opcode that analyzes events to determine if rerating is required. For example, the criteria for rerating might be:

- If the event is a cancel event.
- If the cancel event for a charge offer requires proration on cancellation.
- If rerating needs to occur to calculate proration for this charge offer.

If the notification event triggers rerating, the custom opcode specifies which events for an account must be rerated and sends the rerating requirements to the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode to create a rerate job.

Trigger-dependent rerating works as follows:

1. An event that you have configured in event notification occurs to call the custom opcode. For example, this might be a purchase event, charge offer cancellation, or change in account status. All the fields in the event are passed as input to the custom opcode.
2. The custom opcode analyzes the event using your custom selection criteria to determine if it should trigger rerating. For example, the opcode might be called when all purchase events occur, but not all purchase events will trigger rerating.
3. If the event triggers rerating because it matches your selection criteria, the custom opcode assigns an optional rerate reason code for rerating those events, specifies any price overrides, and calls PCM_OP_RERATE_INSERT_RERATE_REQUEST to create the rerate job.
4. PCM_OP_RERATE_INSERT_RERATE_REQUEST creates a rerate job that includes:
   - The account that needs to be rerated.
   - The events that must be rerated for that account.

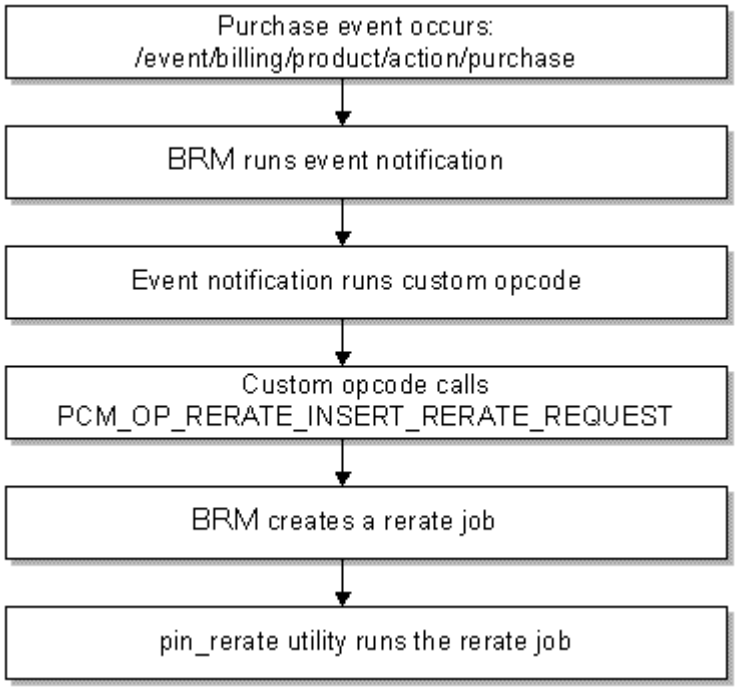
- The start time from which to rate the events.
- The rerate reason.
- Price overrides, if any.

> **Note:**
>
> If more than one account is included in the rerate job, the same selection criteria, price overrides, and start time apply to all the accounts.

5. You run the **pin_rerate** utility to process the rerate job and rerate the events.

Figure 7-2 shows the trigger-dependent rerating process when a purchase event occurs.

**Figure 7-2 Trigger-Dependent Rerating Process**




To set up trigger-dependent rerating, see "Setting Up Trigger-Dependent Rerating".

# Setting Up Trigger-Dependent Rerating

To set up trigger-dependent rerating, you do the following:

1. Create a custom opcode for trigger-dependent rerating. See "Creating a Custom Opcode for Trigger-Dependent Rerating".
2. Configure event notification for trigger-dependent rerating. See "Configuring Event Notification for Trigger-Dependent Rerating".

## Creating a Custom Opcode for Trigger-Dependent Rerating

To use trigger-dependent rerating, you must write custom code to specify when to create rerate jobs automatically when certain events occur. You can write your code by doing one of the following:

- Create one or more custom opcodes.

  You can create multiple custom opcodes to handle rerating events for different scenarios or create one opcode to handle all scenarios.

- Use the PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST policy opcode.

  This policy opcode handles rerating events for BRM automatic rerating scenarios. You can modify and recompile this policy opcode to handle your own automatic rerating scenarios.

The opcode you use to define your custom code for trigger-dependent rerating must call the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode.

Your custom code is required to pass in the following to PCM_OP_RERATE_INSERT_RERATE_REQUEST:

- The POIDs of the accounts to be rerated.

  You can also pass in an optional array of additional accounts that are related to these accounts (for example, an account that used the account's service before a line transfer) and the start time for each account.

  > **Note:**
  >
  > If an array of accounts is passed in, the same selection criteria and price overrides apply to all the accounts.

- The time from which events must be rerated for the accounts.

Your custom code can optionally pass in the following to PCM_OP_RERATE_INSERT_RERATE_REQUEST:

- A rerate reason code to take advantage of rerating according to type of rerate job. See "Specifying a Rerate Reason Code".
- Selection criteria so that only those events that are required for rerating are identified for an account. See "Specifying Selection Criteria".
- Price overrides to substitute an account's subscribed pricing for alternate pricing during rerating. See "Specifying Price Overrides".

Your custom selection criteria and price override information for creating rerate jobs can include anything that can be passed as input to the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode. For detailed information about the array and field names that can be passed, refer to the PCM_OP_RERATE_INSERT_RERATE_REQUEST input and output flist specifications and the class specifications of the **/job/rerate** and **/job_batch/rerate** storable classes.

When an event triggers your custom opcode, your opcode may need to obtain data from other events in the database to obtain all of the information required for rerating.

## Specifying a Rerate Reason Code

PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST or your custom opcode can pass an optional rerate reason code to PCM_OP_RERATE_INSERT_RERATE_REQUEST. The rerate reason code is stored in the rerate job and can be used to select jobs from the rerate queue for rerating using **pin_rerate**.

Your rerate reason codes can be any integer, except 1 (1 is a reserved default value for pipeline-generated rerating requests).

> **Note:**
>
> Because you can use multiple rerate reason codes to group rerate jobs for different rerating scenarios, ensure that your rerate reason codes are unique.

The default reason code is **0**.

## Specifying Selection Criteria

When an account needs to be rerated, not every event for that account may need to be rerated from the specified rerate start time. Your custom opcode can optionally send selection criteria to the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode so that all events for the account can be filtered to select only the subset of events that needs to be rerated.

For example, you may need to rerate events only when they are generated by a specific service or only when they are rated by a specific charge offer or discount offer. You can specify selection criteria to rerate all accounts' events that have an event type of **/event/delayed/session/telco/gsm/telephony** but only when they are for the service type **/service/telco/gsm/telephony** and only when they are rated by the charge offer GSM Voice Telephony.

Your selection criteria can be any of the following:

- Event types (array PIN_FLD_EVENTS)
- Service types (array PIN_FLD_SERVICES)
- Charge offer POIDs (array PIN_FLD_PRODUCTS)
- Discount POIDs (array PIN_FLD_DISCOUNTS)
- Bundle POIDs (array PIN_FLD_DEALS)

> **Note:**
>
> Charge offers and discount offers are mutually exclusive with bundles. Bundles include charge offers and discount offers, so you can pass in a bundle array instead of the charge offer array and discount array.

## Specifying Price Overrides

You can specify charge offers, discount offers, or bundles to use for a specific rerate job to override an account's subscribed charge offer, discount offer, or bundles during rerating. The

price override applies only to that rerate job; subsequent rerate jobs use the subscribed pricing or their own override pricing.

Your custom opcode can send the following price override data to the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode to set a price override for a rerate job:

- An optional array of override charge offer POIDs along with the charge offer it is overriding (PIN_FLD_OVERRIDE_PRODUCTS).
- An optional array of override discount POIDs along with the discount it is overriding (PIN_FLD_OVERRIDE_DISCOUNTS).
- An optional array of override bundle POIDs along with the bundle it is overriding (PIN_FLD_OVERRIDE_DEALS).

> **Note:**
>
> Charge offers and discount offers are mutually exclusive with bundles because bundles include charge offers and discount offers. The bundle is translated to a list of charge offers and discount offers when it is passed to the rerating opcodes.

The override charge offers and discount offers or bundles must already be defined in your BRM database. They must also be functionally equivalent to the subscribed charge offers and discount offers or bundles; for example, the priority or event map would be the same. In addition, override charge offers and discount offers cannot be configured as first usage charge offers.

> **Note:**
>
> You must rerate accounts when an override pricing charge offer is canceled in a given billing cycle so that refunds can be applied correctly. See "Configuring Event Notification for Override Pricing".

BRM creates a record when rerating is run with price overrides by using the **/rerate_session/override_info** object. When override pricing is passed to the PCM_OP_SUBSCRIPTION_RERATE_REBILL opcode, this opcodes creates a **/rerate_session/override_info** object to capture the override pricing information used during rerating.

Price overrides are automatic when you use the Best Pricing feature. For that feature, BRM calculates the best price from alternate charge offers and discount offers and rerates events to use the best pricing. When rerating events, the charge offer or discount offer that provides the best pricing is the override charge offer or discount offer.

## Configuring Event Notification for Trigger-Dependent Rerating

When you configure event notification for trigger-dependent rerating, you specify the following in the **pin_notify** file in *BRM_home***/sys/data/config** (or your own merged event notification configuration file):

- The events that trigger rerating.

- The custom opcodes you want BRM to use to analyze the events, to identify whether rerating is required.

To configure event notification for trigger-dependent rerating, do the following:

1. If your system has multiple configuration files for event notification, merge them.
2. Ensure that the merged file includes the events you want to trigger rerating and the opcode number of the custom opcode you want to analyze those events.

   For example, these entries call the custom opcode with opcode number 10000:

   ```
   # Event notification for trigger-dependent rerating
   10000    0    /event/customer/status
   10000    0    /event/billing/product/action/purchase
   ```

3. (Optional) Add, modify, or delete entries in your final event notification list.
4. (Optional) Create custom code for event notification to trigger.
5. Load your final event notification list into the BRM database.

## Configuring Event Notification for Override Pricing

If you use override pricing, you must set up trigger-dependent rerating to rerate accounts for cases where a refund could not be applied because an override pricing charge offer was canceled in a given billing cycle.

For a given billing cycle, if you rerate an account with an override charge offer and the charge offer is canceled, BRM cannot calculate a refund for that account because the charge offer is not available to apply the necessary cycle fees. When the refund cannot be applied, BRM creates the notification event **/event/notification/product/cancel/no_refund**.

To calculate the correct refund amount, you must set up trigger-dependent rerating as follows:

1. Write custom code to specify:
   - The account to rerate from the **/event/notification/product/cancel/no_refund** event.
   - The charge offer to use to apply the cycle fees for the refund to use the override charge offer that was canceled.

     BRM uses the base pricing charge offer associated with the account if you do not specify the override charge offer that was canceled. To specify the override charge offer that was canceled, obtain its information from the latest **/rerate_session/override_info** object created for the given account.
2. Include your code in a custom opcode or in the PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST policy opcode that handles automatic rerating (opcode number 3787).

   If you create a custom opcode, it must call the PCM_OP_RERATE_INSERT_RERATE_REQUEST opcode to create the rerate job.
3. Set up event notification for the event **/event/notification/product/cancel/no_refund** to call your custom opcode or the policy opcode. For example, to call the PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST policy opcode, you would enter:

   ```
   # Event notification for trigger-dependent rerating
   3787    0    /event/notification/product/cancel/no_refund
   ```

   For detailed instructions on setting up event notification, see "Configuring Event Notification for Trigger-Dependent Rerating".

8

# Using the pin_rerate Utility

You use the Oracle Communications Billing and Revenue Management (BRM) **pin_rerate** utility to select and rerate accounts.

Topics in this document:



## About Using the pin_rerate Utility

You use the **pin_rerate** command-line utility to perform the following tasks:

- Select accounts and events for rerating from the BRM database. When accounts are selected, **pin_rerate** creates rerate jobs for the selected accounts. See "Selecting Accounts and Events for Rerating".

  You can assign a rerate reason code to the jobs created for the selected accounts and events. This enables you to rerate the accounts separately based on the reason for rerating. See "Assigning Rerate Reasons to Rerate Jobs".

  You can also define custom **pin_rerate** parameters based on various event criteria. This enables you to further customize which event attributes are used to select accounts and events for rerating. See "Defining Custom pin_rerate Parameters for Rerating".

- Rerate the accounts. To rerate accounts, you process rerate jobs. The process for rerating depends on whether you rerate real-time-rated and pipeline-batch-rated events concurrently or, if you do not use Pipeline Manager for batch rating, rerate only real-time-rated events. See the following sections:

  – Rerating Real-Time-Rated and Pipeline-Rated Events Concurrently

  – Processing Rerate Jobs When You Do Not Use Pipeline Batch Rating

  For information about processing jobs that are automatically created by BRM automatic rerating features, see "About Processing Rerate Jobs Created by Automatic Rerating".

> **Note:**
>
> Do not move accounts to another database schema while rerating events for those accounts.

- Reprocess any rerate jobs that failed. See "Processing Failed Rerate Jobs".
- Back out the balance impacts of rating without rerating the events. See "Using pin_rerate for Back-Out-Only Rerating".
- Generate reports about the results of rerating. See "Reports Generated by the pin_rerate Utility".

If rerating fails, **pin_rerate** creates a report that includes the account numbers and start times for failed rerate jobs. The report file name is **pin_rerate.status_report** and is in the directory from which you ran the utility.

When rerate jobs have been processed, you can run **pin_rerate** to purge them from the database.

# Selecting Accounts and Events for Rerating

The first step in the rerating process is running **pin_rerate** to select the accounts and associated events to rerate from the BRM database. The **pin_rerate** utility creates rerate jobs (**/job/rerate** and **/job_batch/rerate** objects) to store the information about the selected accounts.

> **Note:**
>
> - If you use automatic rerating, accounts are selected and rerate jobs are created automatically for certain scenarios.
> - Do not move accounts to another database schema while rerating events for those accounts.

## Specifying Accounts for Rerating

By default, the **pin_rerate** utility selects for rerating all the accounts and then all the events associated with those accounts that occurred from the start time that you specify.

For example, the following command selects all the accounts from the BRM database and, for those accounts, selects all the events that occurred after 07/23/2025.

```
pin_rerate -t 07/23/2025
```

**pin_rerate** provides a set of parameters that you can optionally use to select only specific accounts that meet one of the following requirements:

- One or a set of accounts identified by the account POIDs
- Accounts with events rated by a particular charge offer
- Accounts with events rated by a particular discount offer

- Accounts with events rated by charge offers and discount offers associated with a particular bundle
- Accounts with events generated for a particular service type or subscription service
- Accounts with events associated with an account's bill unit or bill unit and balance group
- Accounts that have particular event types

For example, when you use the **-p** parameter:

```
pin_rerate -p products.txt -t 07/23/2025
```

**pin_rerate** does the following:

- Selects only the accounts that have events related to the charge offers in the **products.txt** file.
- Selects all the events for the selected accounts that occurred after 07/23/2025.

When you use the **-line** parameter:

```
pin_rerate -line 6495832245 -t 09/21/2024
```

**pin_rerate** does the following:

- Selects only the accounts that have the subscription service with the phone number 6495832245.
- Selects all the events for the selected accounts that occurred after 09/21/2024.

## Specifying Events for Rerating

By default, **pin_rerate** rerates all the events for the selected accounts from the start time that you specify. You can specify to rerate only specific events for the selected accounts by using the **-r** parameter. This is called *selective rerating*.

> **Note:**
>
> Do not use selective rerating if:
>
> - Your rating scheme includes credit limits or balance element-based tiers. These schemes require that all events related to an account are rerated to assure accurate rerating.
> - Deferred taxation is used for taxing events during rerating.

If you use selective rerating, be sure to consider how it might affect rating overall because account balances can be impacted by different types of events: a cycle event can grant free minutes and a usage event consumes free minutes from the same balance. If, for example, you change the pricing for a charge offer that grants free minutes, you must rerate all events for the accounts that own the charge offer. It would be incorrect to use selective rerating in this case.

The **-r** parameter must be used with parameters that specify the accounts to select for rerating. When the **-r** parameter is used, rerating applies the account selection criteria to the account's events as well and selects only those events that meet the selection criteria.

The **-r** parameter can be used with any of the account selection parameters.

For example, when you use **-r** with the **-s** parameter:

```
pin_rerate -r -s service.txt -t 07/23/2025
```

**pin_rerate** does the following:

- Selects only the accounts that have events related to the services in the **service.txt** file.
- Selects only the events related to the services in **service.txt** that occurred after 07/23/2025 for the selected accounts.

When you have a high volume of events to rerate, you can rerate events that are rated only in real time, such as cycle and purchase events. To do this, you use the **-r** parameter in combination with the **-n** parameter for specifying event types. You define all the event types rated by real-time rating in an input file.

For example, you could specify the following event types in a file named **event.txt**:

- **/event/billing/product/fee/cycle/cycle_forward_monthly**
- **/event/billing/product/fee/purchase**
- **/event/billing/product/fee/cycle/cycle_forward_arrear**

When you run the following command:

```
pin_rerate -t 01/01/2025 -n event.txt -r
```

**pin_rerate** does the following:

- Selects all accounts that have events with event types in the **event.txt** file.
- Selects only the events with event types in the **event.txt** file and which occurred after 01/01/2025 for the selected accounts.

When rerating cycle fee events, to get the correct rerating results, include the cycle events that occur during billing that are configured in the charge offer, such as cycle discount, rollover, and fold events in the event file.

For example, if a cycle discount is configured to be some percentage of the charge during billing and if the cycle forward arrear fee is modified during the billing cycle, then to rerate the cycle forward arrear event, you need to include both events in the event file:

- **/event/billing/product/fee/cycle/cycle_forward_arrear**
- **/event/billing/cycle/discount**

If the event type specified in the **-n** parameter input file has subclasses, all subclass events are also rerated, providing they meet the selection criteria. For example, if you specify **/event/delayed/session/telco** in the **-n** parameter input file, events of type **/event/delayed/session/telco/gsm** that meet the selection criteria are also rerated.

## Customizing Event Searches for Selective Rerating

You can further refine the event selection criteria used for selective rerating by customizing the PCM_OP_SUBSCRIPTION_POL_SPEC_RERATE policy opcode. This opcode is called when the **pin_rerate** utility is run with **-r** parameter to indicate selective rerating.

> **Note:**
>
> An alternative to customizing this policy opcode to filter the events selected for rerating is to create custom **pin_rerate** parameters instead. See "Defining Custom pin_rerate Parameters for Rerating".

PCM_OP_SUBSCRIPTION_POL_SPEC_RERATE receives an event search template that is based on the account and event selection criteria specified in the **pin_rerate** command line: for example, to select events related to services specified by the **-s** parameter.

By default, PCM_OP_SUBSCRIPTION_POL_SPEC_RERATE does not change the search template and returns a copy of the input flist in the output flist.

You can customize the search template in the input flist to rerate specific types of events. Most customizations include changes only to the fields listed in Table 8-1.

**Table 8-1 Fields to Customize**

| Field | Description |
|---|---|
| PIN_FLD_TEMPLATE | The modified search template. |
| PIN_FLD_ARGS | The list of search arguments.<br>**Note:**<br>• This list must match the list of arguments in PIN_FLD_TEMPLATE.<br>• It is preferable to have arguments of one type only. For example, an event search based on charge offer objects. |

The PCM_OP_SUBSCRIPTION_POL_SPEC_RERATE policy opcode receives the following fields in the RESULTS array from the PCM_OP_SUBSCRIPTION_RERATE_REBILL standard opcode:

- PIN_FLD_POID
- PIN_FLD_CREATED_T
- PIN_FLD_EFFECTIVE_T
- PIN_FLD_END_T
- PIN_FLD_SERVICE_OBJ
- PIN_FLD_ACCOUNT_OBJ
- PIN_FLD_UNRATED_QUANTITY
- PIN_FLD_RERATE_OBJ
- PIN_FLD_BAL_IMPACTS [*]
- PIN_FLD_SUB_BAL_IMPACTS [*]

> **Note:**
>
> - To assure that the existing mandatory fields in the array are passed back, avoid customizing the RESULTS array. Extra fields added to the array are retrieved but ignored by the standard opcode.
> - Test your customized template to ensure that it works properly before using it with a working database. Use the **-e** and **-c** parameters with the **pin_rerate** utility to test your modifications.

The PCM_OP_SUBSCRIPTION_POL_SPEC_RERATE policy opcode returns the search template that the PCM_OP_SUBSCRIPTION_RERATE_REBILL opcode uses to find events in selected accounts that need rerating.

## Specifying the Event Sequence for Rerating

Events are rerated in sequence based on the event time. You can specify one of two times:

- *Event end time* defines the time that the usage actually occurred. This is the default.
- *Event creation time* is the time that the event was loaded into the BRM database.

The event time you specify might depend on how the original events were rated:

- **Events rated or loaded in batches**

  Events that are rated or loaded into the BRM database in batches have a significant delay between the event end time and creation time. Using the event end time reflects the actual real-time sequence of the original events. However, because batch events are recorded in order of creation time, this makes predicting the actual impact of a price configuration change harder. To compare the original and rerated balance impacts of batch events, use the event creation time.

- **Events rated and loaded in real time**

  Events that are rated and loaded in real-time have very little delay between the event end time and creation time. Both the event end time and creation time reflect the real-time sequence in which the original events occurred and were recorded.

By default, the **pin_rerate** utility rerates batch events in sequence based on event end time for both real-time and batch events.

To specify the event time for rerating, use the **-b** parameter when selecting accounts for rerating. The **-b** parameter takes either the **c** option (for event creation time) or the **e** option (for event end time).

For example:

```
pin_rerate -b c -t 07/23/2025
```

The preceding example selects all accounts, then selects both real-time and batch events that occurred after 07/23/2025 for the selected accounts, and finally rerates the events in order of creation time.

# Assigning Rerate Reasons to Rerate Jobs

Some rerating jobs must be processed before others. For example, if a rerate job includes events that have an impact on future rating, such as volume-based discounting, those events

must be rerated first. You can achieve this by assigning rerate reason codes to rerate jobs when you create those jobs. You can then select only those jobs associated with the specified rerate reason code during scheduled rerating process.

Rerate reason codes can be any integer except **1** (**1** is reserved for pipeline-generated events that were rated out of order).

> **Note:**
>
> Ensure that your reason codes are unique to process rerate jobs associated with them during separate rerating process.

You assign rerate reason codes when you manually create rerate jobs by using the **pin_rerate** utility.

> **Note:**
>
> Rerate jobs created automatically by BRM automatic rerating features are not assigned a reason code by default. You can assign rerate reason codes for trigger-dependent and automatic rerating by customizing the PCM_OP_SUBSCRIPTION_POL_GENERATE_RERATE_REQUEST policy opcode.
>
> You can assign a unique reason code for out-of-order rerating by using a configuration file.

To assign a rerate reason code by using **pin_rerate**, you specify the rerate reason code on the command line along with other account selection parameters, such as the start time and other parameters:

```
pin_rerate -reason reason_code -t start_time other_parameters
```

You can specify only one reason code when creating rerate jobs. Rerate jobs are created for the selected accounts and all rerate jobs are assigned the specified rerate reason code.

For example, to assign a rerate reason code of **99** to rerate jobs that rerate all account's events that occurred after January 1, 2025 and that are associated with the charge offers specified in the **product.txt** file, you enter the following command:

```
pin_rerate -reason 99 -t 01/01/2025 -p product.txt
```

# Defining Custom pin_rerate Parameters for Rerating

You can define custom **pin_rerate** parameters based on any event criteria. This enables you to customize which event attributes are used to select accounts and events for rerating.

To define custom parameters, you map extraction keys to fields in the event object. You then run the **load_pin_rerate_flds** utility to load the extraction-key-to-event-field mappings into the **/config/rerate_flds** object in the BRM database. See "Configuring Custom pin_rerate Parameters".

When you run **pin_rerate**, it uses the extraction key to find the corresponding event field name in the **/config/rerate_flds** object. It uses the event field name to find and retrieve accounts and events for rerating.

For example, if you map the extraction key **resource_id** to the PIN_FLD_RESOURCE_ID field in the **/event** object, you can run **pin_rerate** with the following command, specifying the input file containing the balance element IDs:

```
pin_rerate -resource_id input_file -t start_time
```

In this example, **pin_rerate** selects accounts to be rerated that have events associated with the balance elements specified in the input file.

By default, BRM searches only the base **/event** class for fields associated with custom **pin_rerate** parameters. If the event field associated with a custom parameter is present only in a subclass, you must specify the subclass event type in the command line by including the **-n** parameter.

> **Note:**
>
> When used with custom **pin_rerate** parameters, the input file for the **-n** parameter can contain only one event type. If you specify more than one event type, an error is returned.

# Configuring Custom pin_rerate Parameters

To configure custom **pin_rerate** parameters, you perform the following tasks:

- Defining Custom Parameters
- Loading Custom Parameters

## Defining Custom Parameters

To define custom **pin_rerate** parameters, create an XML file that maps the parameters to event fields. You can map to fields specified in the EVENT_T and EVENT_BAL_IMPACTS_T tables in the base **/event** class or to fields in an event subclass.

> **Note:**
>
> - If you map parameters to fields in an **/event** subclass, you must specify the subclass by using the **-n** parameter when you run **pin_rerate** with the custom parameter.
> - To create the XML file, you must be familiar with XML and the XML schema.

Create an XML file containing the parameter-to-event field mappings according to the BRM rerating XML schema file (*BRM_home***/xsd/pin_rerate_flds.xsd**).

> **Note:**
>
> Validate your XML file against the XML schema. The **load_pin_rerate_flds** utility cannot successfully load an invalid XML file.

### Sample XML parameter file

The following example shows how you specify event fields in **<Name>** tags and the parameters they map to in **<value>** tags:

```
<PinRerateList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.portal.com/InfranetXMLSchema pin_rerate_flds.xsd">
     <PinRerate>
          <Name>EVENT.PIN_FLD_START_T </Name>
          <value>startTime</value>
     </PinRerate>
     <PinRerate>
          <Name>EVENT.PIN_FLD_END_T</Name>
          <value>endTime</value>
     </PinRerate>
     <PinRerate>
          <Name>EVENT.PIN_FLD_ACCOUNT_OBJ</Name>
          <value>account</value>
     </PinRerate>
     <PinRerate>
          <Name>EVENT.PIN_FLD_RATE_PLAN </Name>
          <value>ratePlan</value>
     </PinRerate>
     <PinRerate>
          <Name>EVENT.PIN_FLD_DEAL </Name>
          <value>deal</value>
     </PinRerate>
     <PinRerate>
          <Name>EVENT.PIN_FLD_SERVICE_OBJ </Name>
          <value>service</value>
     </PinRerate>
     <PinRerate>
          <Name>EVENT.PIN_FLD_BAL_IMPACTS.PIN_FLD_PRODUCT_OBJ</Name>
          <value>product</value>
     </PinRerate>
</PinRerateList>
```

The following list provides information and examples of how you specify event fields in **<Name>** tags and the parameters they map to in **<value>** tags:

1. The <value> tag in EVENT.PIN_FLD_START_T and EVENT.PIN_FLD_END_T accept input in the format SS/MM/HH/MM/DD/YYYY, where:
   - SS is the seconds (0 – 59).
   - mm is the minute of the hour (0 – 59).
   - HH is the hour of the day (0 – 23).
   - MM is months in digits.
   - DD is the two-digit day of the month.
   - YYYY is the 4-digit year.

   For example: 45/09/06/05/01/2024
2. For EVENT.PIN_FLD_BAL_IMPACTS.PIN_FLD_PRODUCT_OBJ, you input a file that contains the product name.

   If you use an out-of-the-box product as an example, the file will contain Product 1b - Email Account as the product name.

   However, you can use product POID if the custom event name is EVENT.PIN_FLD_BAL_IMPACTS.PRODUCT_OBJ.

3. To use object POID for the following event names, remove PIN_FLD_.

   For example:

   - ACCOUNT_OBJ
   - DISCOUNT_INFO
   - DEAL
   - SERVICE_OBJ
   - PRODUCT_OBJ
   - RATE_PLAN

   The following example shows how you specify event fields in <Name> tags:

   ```
   <PinRerate>
            <Name>EVENT.ACCOUNT_OBJ</NAME>
            <value>account</value>
   </PinRerate>
   ```

   For object POID, use a shortened version of the event name in <value>product</value>.

   For example:

   - account
   - discount or discountInfo
   - deal
   - service
   - product

4. If you map a custom parameter to a field in an **/event** subclass, specify the array or substruct (if any) that contains the parameter field.

   For example, to specify PIN_FLD_ORIGIN_NETWORK, which is located in the PIN_FLD_TELCO_INFO substruct in the **/event/delayed/session/telco** subclass:

   ```
   <PinRerate>
   <Name>EVENT.PIN_FLD_TELCO_INFO.PIN_FLD_ORIGIN_NETWORK</Name>
   <value>origin_network</value>.
   </PinRerate>
   ```

## Loading Custom Parameters

Use the following command to load the custom **pin_rerate** parameters defined in the XML file into the **/config/rerate_fld** object:

*BRM_home*/**sys/data/config/load_pin_rerate_flds** *xml_file_name*

> **Note:**
>
> The **load_pin_rerate_flds** utility uses a configuration file (**pin.conf**) located in the same directory to connect to the BRM database. Edit the configuration file to connect to your BRM database.

# About Processing Rerate Jobs Created by Automatic Rerating

Rerate jobs are created automatically by the following features:

- Automatic rerating.
- Trigger-dependent rerating.
- Out-of-order rerating.

These rerate jobs are automatically processed when you run the **pin_rerate** with one of the following parameters:

- The **-process jobs** parameter when rerating real-time-rated and pipeline-rated events concurrently.
- The **-rerate** parameter when rerating only real-time-rated if you do not use Pipeline Manager for batch rating.

By default, rerate jobs that are automatically created are not assigned a rerate reason code. If you customize automatic rerating to assign a specific rerate reason code to rerate jobs, you process those jobs by using the **-reason** parameter with the **-process jobs** or **-rerate** parameter.

For more information, see the following sections:

- Rerating Real-Time-Rated and Pipeline-Rated Events Concurrently
- Processing Rerate Jobs When You Do Not Use Pipeline Batch Rating

# Rerating Real-Time-Rated and Pipeline-Rated Events Concurrently

When rerating real-time-rated and pipeline-rated events concurrently, you process rerate jobs in the following ways:

- By processing all existing rerate jobs that were previously created. See "Processing Rerate Jobs for Concurrent Rerating".
- By processing only rerate jobs associated with a rerate reason code. See "Processing Rerate Jobs According to a Rerate Reason Code for Concurrent Rerating".

> **Note:**
>
> Ensure that no event data records (EDRs) are in the pipeline before you run **pin_rerate**. Otherwise, EDRs might be rated using old account data.

## Processing Rerate Jobs for Concurrent Rerating

When rerating real-time-rated and pipeline-batch-rated events concurrently, rerate jobs are created and processed in separate commands. Rerate jobs can be created in the following ways:

- Automatically by BRM automatic rerating features.

- When you run **pin_rerate** to select accounts for rerating, as described in "Selecting Accounts and Events for Rerating". For example:

  ```
  pin_rerate -t start_time -s input_file
  ```

To process all existing rerate jobs in the rerate queue, run the following commands:

1. **pin_rerate -process jobs**. See "Notifying the Batch Pipeline That Rerating Is Starting".
2. **pin_rerate -process queue**. See "Processing Acknowledgment Events from the Batch Pipeline".
3. **pin_rerate -rerate**. See "Rerating the Events Associated with the Accounts in Rerate Jobs".
4. **pin_rerate -process queue**. See "Processing Acknowledgment Events from the Batch Pipeline".
5. **pin_rerate -process recycle**. See "Recycling EDRs Suspended During Rerating".

For example:

```
pin_rerate -process jobs
pin_rerate -process queue
pin_rerate -rerate
pin_rerate -process queue
pin_rerate -process recycle
```

> **Note:**
>
> You run **pin_rerate** during non-peak hours with these commands as part of a cron job. See your operating system documentation for details on creating a cron job.

## Notifying the Batch Pipeline That Rerating Is Starting

After rerate jobs are created, you run **pin_rerate** with the **-process jobs** parameter to notify the batch pipeline that rerating is about to begin.

When you specify the **-process jobs** parameter, the following actions are performed:

1. The **pin_rerate** utility finds all the rerate jobs with a status of NEW and sends business events containing the rerate job ID and the list of accounts associated with the rerate job to the batch pipeline.
2. The **pin_rerate** utility then changes the status of the jobs from NEW to WAITING_ACCOUNT_LOCKED to indicate that it is waiting for an acknowledgment from the pipeline.
3. The batch pipeline dequeues the business events and suspends rating EDRs for those accounts.
4. The batch pipeline sends an acknowledgment event back to the AQ database queue.

> **Note:**
>
> - If you process rerate jobs according to a rerate reason, the same actions are performed but only for the rerate jobs associated with the rerate reasons you specify.
> - To process rerate jobs according to a rerate reason by using the **pin_rerate** utility, see "Processing Rerate Jobs According to a Rerate Reason Code for Concurrent Rerating".

## Processing Acknowledgment Events from the Batch Pipeline

You run **pin_rerate** with the **-process queue** parameter to process acknowledgment events sent by the batch pipeline.

You run **pin_rerate** with the **-process queue** parameter at two times:

- After you run **pin_rerate** with the **-process jobs** parameter. The acknowledgment event at this time indicates that the batch pipeline has suspended rating EDRs for the accounts in the rerate jobs. The **pin_rerate** utility dequeues the event and changes the status of the rerate jobs from WAITING_ACCOUNT_LOCKED to ACCOUNT_LOCKED.
- After you run **pin_rerate** with the **-rerate** parameter. The acknowledgment event at this time indicates that the batch pipeline has resumed rating EDRs for the accounts in the rerate job. **pin_rerate** dequeues the event and changes the status of the rerate jobs from RERATED to READY_FOR_RECYCLE.

## Rerating the Events Associated with the Accounts in Rerate Jobs

When the rerate jobs has the ACCOUNT_LOCKED status, you run **pin_rerate** with the **-rerate** parameter to rerate the accounts.

**pin_rerate** finds all the rerate jobs with a status of ACCOUNT_LOCKED and calls rerating opcodes to rerate the accounts.

After an account is successfully rerated, **pin_rerate** sends a business event through the Oracle Data Manager (DM) to the batch pipeline to update the account data. The DAT_BalanceBatch module processes the event and reloads the updated account balances from the BRM database.

When *all* accounts in a rerate job have been successfully rerated and updated, **pin_rerate** updates the rerate job status from ACCOUNT_LOCKED to RERATED and then notifies the batch pipeline that rerating is complete. The batch pipeline resets the rerate flags for all the accounts in the rerate job and returns an acknowledgment event to inform **pin_rerate** that it has resumed rating EDRs for those accounts.

## Recycling EDRs Suspended During Rerating

EDRs that are temporarily suspended by the batch pipeline are loaded into the BRM database by Suspended Event (SE) Loader. The suspended EDRs are stored in the database until they are recycled.

> **Note:**
>
> Before you can recycle suspended EDRs, they must be loaded into the BRM database by SE Loader. You typically schedule SE Loader to run automatically when you set up standard recycling.

To recycle the EDRs that were suspended during the rerating process, you run **pin_rerate** with the **-process recycle** parameter.

> **Note:**
>
> Suspended EDRs are typically recycled by running the **pin_recycle** utility. However, **pin_rerate** calls the standard recycling opcode directly so you do not use **pin_recycle** when using **pin_rerate**.

**pin_rerate** finds all the rerate jobs with a status of READY_FOR_RECYCLE and calls the standard recycling opcodes to recycle the associated EDRs. Standard recycling uses the recycle key value in the EDR to identify and retrieve the EDRs that were suspended during the rerating process.

After the EDRs are recycled, **pin_rerate** changes the status of the jobs from READY_FOR_RECYCLE to COMPLETE.

> **Note:**
>
> - In a multischema environment, you must run **pin_rerate** separately on each database schema.
> - If no EDRs were suspended for the accounts being rerated, the job status is still changed to COMPLETE.
> - If an error occurs while recycling EDRs, the job status is not changed; it retains the status of READY_FOR_RECYCLE.

# Processing Rerate Jobs According to a Rerate Reason Code for Concurrent Rerating

To process rerate jobs according to a rerate reason code when rerating real-time-rated and pipeline-batch-rated events concurrently, you use the **-process jobs** parameter with the **-reason** parameter:

```
pin_rerate -process jobs -reason [reason_code_1,reason_code_2,reason_code_3,...]
```

where *reason_code_x* is the rerate reason code assigned to existing rerate jobs in one of the following ways:

- When you run **pin_rerate** with the **-reason** parameter to create rerate jobs and with an assigned rerate reason code. See "Assigning Rerate Reasons to Rerate Jobs".

- When you customize one of the following features to assign a specific rerate reason code to rerate jobs that are automatically created:
  - Automatic rerating.
  - Trigger-dependent rerating.
  - Out-of-order rerating.

You can list multiple reason codes separated by commas (do not use spaces between reason codes). For example, to process jobs with reason codes 100, 200, and 300, enter the following commands:

```
pin_rerate -process jobs –reason 100,200,300
pin_rerate -process queue
pin_rerate -rerate
pin_rerate -process queue
pin_rerate -process recycle
```

> **Note:**
>
> When you list multiple rerate reasons, the rerate jobs associated with them are processed randomly. They are not processed in the order you list them in the command line, nor are they processed in increasing or decreasing numeric order. To process rerate jobs according to a rerate reason in a particular order, you must schedule separate rerating process for them.

# Processing Rerate Jobs When You Do Not Use Pipeline Batch Rating

If you do not use Pipeline Manager for batch rating, you process rerate jobs in the following ways:

- By creating rerate jobs and processing those jobs at the same time. See "Processing Rerate Jobs When Selecting Accounts for Rerating".
- By processing all existing rerate jobs that were previously created. See "Processing Existing Rerate Jobs".
- By processing only rerate jobs associated with a rerate reason code. See "Processing Rerate Jobs per a Rerate Reason When Rerating Only Real-Time-Rated Events".

## Processing Rerate Jobs When Selecting Accounts for Rerating

When rerating real-time events only, rerate jobs are processed when you run **pin_rerate** and specify the start time and any additional search criteria as described in "Selecting Accounts and Events for Rerating".

For example:

```
pin_rerate -t start_time -p input_file
```

The **pin_rerate** utility selects the accounts for rerating by using the search criteria, creates rerate jobs, and then immediately rerates the selected accounts.

## Processing Existing Rerate Jobs

Some rerate jobs are automatically created by BRM automatic rerating features. Other rerate jobs can be precreated by assigning a rerate reason code when selecting the accounts for rerating.

To process all existing rerate jobs in the rerate queue, you use only the **-rerate** parameter:

```
pin_rerate -rerate
```

## Processing Rerate Jobs per a Rerate Reason When Rerating Only Real-Time-Rated Events

To process rerate jobs according to a rerate reason when rerating only real-time-rated events, you use the **-rerate** parameter with the **-reason** parameter:

```
pin_rerate -rerate -reason reason_code_1,reason_code_2,reason_code_3,...
```

where *reason_code_x* is the rerate reason code assigned to existing rerate jobs in one of the following ways:

- When you run **pin_rerate** with the **-reason** parameter to create rerate jobs with an assigned rerate reason code. See "Assigning Rerate Reasons to Rerate Jobs".
- When you customize one of the following features to assign a specific rerate reason code to rerate jobs that are automatically created:
  - Automatic rerating.
  - Trigger-dependent rerating.
  - Out-of-order rerating.

You can list multiple reason codes separated by commas (do not use spaces between reason codes). For example, to process jobs with reason codes 100, 200, and 300, you enter the following commands:

```
pin_rerate -rerate –reason 100,200,300
```

> **Note:**
>
> When you list multiple rerate reasons, the rerate jobs associated with them are processed randomly. They are not processed in the order you list them in the command line, nor are they processed in increasing or decreasing numeric order. To process rerate jobs according to a rerate reason in a particular order, you must schedule separate rerating process for them.

# Processing Failed Rerate Jobs

To process failed rerate jobs, you run **pin_rerate** only with the commands for processing rerate jobs, without specifying selection criteria or rerate reason codes.

> **Note:**
>
> This also processes all rerate jobs that are in the rerate queue: not only failed rerate jobs.

For example:

- When rerating real-time-rated and pipeline-rated events concurrently, use the following commands:

  ```
  pin_rerate -process jobs
  pin_rerate -process queue
  pin_rerate -rerate
  pin_rerate -process queue
  pin_rerate -process recycle
  ```

  See "Rerating Real-Time-Rated and Pipeline-Rated Events Concurrently".

- If you do not use Pipeline Manager for batch rating, use the following command:

  ```
  pin_rerate -rerate
  ```

  See "Processing Rerate Jobs When You Do Not Use Pipeline Batch Rating".

# Using pin_rerate for Back-Out-Only Rerating

Back-out-only rerating backs out the balance impacts of rating without rerating events.

BRM backs out balance impacts for back-out-only rerating by creating an adjustment event that fully negates the original balance impacts.

To back out the balance impacts of rating, run **pin_rerate** with the **-backout** parameter. Use the **-backout** parameter with other parameters that select the accounts and their events for rerating. This creates rerate jobs that are set for back-out-only rerating for the selected accounts.

For example, the following command selects accounts whose events were rated by the charge offers specified in the **products.txt** file and backs out those events rated by the charge offers that occurred after 06/01/2025:

```
pin_rerate -backout -r -p products.txt -t 06/01/2025
```

The **-r** parameter is used to select only the events that used the specified charge offers. Without the **-r** parameter, **pin_rerate** would select accounts whose events used the specified charge offers and then back out all events for those accounts.

> **Note:**
>
> Use caution when choosing the events to back out because it can impact your general ledger. For example, it is incorrect to use back-out-only rerating for a cycle event when the customer has already paid the cycle fee or to use back-out-only rerating when charge offer pricing is changed. Typically, back-out-only rerating is performed only on usage events where rating should not have occurred.

## Using Custom pin_rerate Parameters with Back-Out-Only Rerating

You can perform back-out-only rerating using custom **pin_rerate** parameters. This enables you to select events for back-out-only rerating based on any event criteria.

For information about defining custom **pin_rerate** parameters, see "Defining Custom pin_rerate Parameters for Rerating".

By default, rerating searches only the base **/event** storable class for fields associated with custom **pin_rerate** parameters. If the event field associated with a custom parameter is present only in a subclass, you must also specify the subclass event type in the command line by using the **-n** *input_file* parameter.

If the event type specified in the **-n** parameter input file has subclasses, all subclass events are also backed out, providing they meet the selection criteria. For example, if you specify **/event/delayed/session/telco** in the **-n** parameter input file, events of type **/event/delayed/session/telco/gsm** that meet the selection criteria are also backed out.

### Example of Using a Custom Parameter with Back-Out-Only Rerating

Suppose you define the custom parameter **origin_network** to select events based on the PIN_FLD_ORIGIN_NETWORK field. You specify an origin network ID in a file named **origin_network.txt**.

The PIN_FLD_ORIGIN_NETWORK field is located in the **/event/delayed/session/telco** subclass so you specify this subclass, in a file named **event_subclass.txt**.

To select accounts whose events were generated by the specified origin network and back out the balance impacts of only those events when they occurred after 06/01/2025, run the following command:

```
pin_rerate -backout -r -n event_subclass.txt -origin_network origin_network.txt -t
06/01/2025
```

# Reports Generated by the pin_rerate Utility

By default, the rerating process generates summary and detailed reports. To print a report, use the **pin_rerate -l** parameter. You have the option of printing either a summary of the report, a detailed report, or both summary and detailed reports. If you do not specify which report to print, both summary and detailed reports are printed by default.

You set the reporting parameter as follows:

- When rerating both real-time-rated and pipeline-batch-rated events concurrently, you specify the **-l** parameter with the **-rerate** parameter. In the following example, the report option is set to print a summary report:

```
pin_rerate -t 01/01/2025 -s service.txt
pin_rerate -process jobs
pin_rerate -process queue
pin_rerate -rerate -l s
pin_rerate -process queue
pin_rerate -process recycle
```

> **Note:**
>
> When the batch pipeline is enabled, you cannot specify the report option at rerate job creation time.

- If you do not use Pipeline Manager for batch rating, you specify the **-l** parameter at one of the following times:
  - When specifying the accounts to rerate. Enter the **-l** parameter after the parameters used to select the accounts. In the following example, the report option is set to print a detailed report:

    ```
    pin_rerate -t 01/01/2025 -p product.txt -l d
    ```

  - When processing rerate jobs that already exist, such as those created with a rerate reason code and those created by automatic rerating. Specify the **-l** parameter with the **-rerate** parameter. In the following example, the report option is set to print a summary report:

    ```
    pin_rerate -rerate -l s
    ```

    > **Note:**
    >
    > Report generation is resource intensive and can degrade system performance, depending on system load and the number of events rerated. You can override report generation by using the **-l n** option with **-rerate** to skip printing reports. For example: **pin_rerate -rerate -l n**.

# Report Generated When Rerating Is Performed Before Billing

Summary reports are created for each account. The following example of a summary report shows that rerating occurred as a result of changing the monthly cycle forward fee from $200 to $20. Rerating took place *before* billing was run, so the difference is shown in the current bill.

```
                PIN_RERATE SUMMARY REPORT
        ========================================

Date Range: 3/1/2025 10:0:0 to 3/2/2025 20:4:49
-------------------------------

Account: 0.0.0.1 /account 13640 0
Resource Name   Original Amount      New Amount      Difference
US Dollar       200.000000           20.000000       -180.000000
Total Resources 200.000000           20.000000       -180.000000
-------------------------------

++++++++++++++++++++++++++++++++++++
The Total Rerate Impact:
++++++++++++++++++++++++++++++++++++
Resource Name   Original Amount      New Amount      Difference
US Dollar       200.000000           20.000000       -180.000000
Total Resources 200.000000           20.000000       -180.000000
-------------------------------------------------
END OF REPORT
================================================================
```

The following is the detailed report for the example above.

```
                PIN_RERATE DETAILED REPORT
          ========================================

Date Range: 3/1/2025 10:0:0 to 3/2/2025 20:4:49

Event: 0.0.0.1 /event/billing/product/fee/cycle/cycle_forward_monthly 219550481834327128
0
Service Type:   /service/ip
Event Time:     3/1/2025 20:4:14
----------
O/N       Resource    Amount       Disct     GL_ID       Tax
Original  US Dollar   -200.000000  0.000      102        0.000000
New       US Dollar    20.000000   0.000      102        0.000000
----------------------------------------
Account: 0.0.0.1 /account 13640 0
Resource Name    Original Amount      New Amount      Difference
US Dollar        200.000000           20.000000       -180.000000
Total Resources  200.000000           20.000000       -180.000000
------------------------------

++++++++++++++++++++++++++++++++++++
The Total Rerate Impact:
++++++++++++++++++++++++++++++++++++
Resource Name     Original Amount      New Amount      Difference
US Dollar         200.000000           20.000000       -180.000000
Total Resources   200.000000           20.000000       -180.000000
-----------------------------------------------
END OF REPORT
```

# Report Generated When Rerating Is Performed After Billing

The following example of a summary report shows rerating as a result of changing the monthly cycle forward fee from $200 to $20. Rerating took place *after* billing was run, so the difference is posted in the *next* bill.

Adjustments are posted only for cycle forward fees. Since cycle forward fees are charged in advance and since the difference between the old and the new amount in this case is $180, the new amount is calculated as -360. In this case, the reports show only the adjusted amount.

```
                PIN_RERATE SUMMARY REPORT
          ========================================

Date Range: 8/7/2025 10:0:0 to 9/7/2025 20:6:52

------------------------------
Account: 0.0.0.1 /account 14854 0
Resource Name     Original Amount      New Amount      Difference
US Dollar         0.000000             -360.000000     -360.000000
Total Resources   0.000000             -360.000000     -360.000000
------------------------------

++++++++++++++++++++++++++++++++++++
The Total Rerate Impact:
++++++++++++++++++++++++++++++++++++
Resource Name      Original Amount      New Amount     Difference
US Dollar          0.000000             -360.000000    -360.000000
Total Resources    0.000000             -360.000000    -360.000000
-----------------------------------------------
END OF REPORT
```

The following is a detailed report of the example above:

```
                PIN_RERATE DETAILED REPORT
        ========================================

Date Range: 8/7/2025 10:0:0 to 9/7/2025 20:6:52


Event:0.0.0.1 /event/billing/adjustment/event 222875404996720238 0
Adjusted From:0.0.0.1 /event/billing/product/fee/cycle/cycle_forward_monthly
222330047229342470 0
Service Type:   /service/ip
Event Time:     9/7/2025 20:6:54
----------
Resource      Amount         Disct     GL_ID
US Dollar     -180.000000    0.000     102
----------

Event:0.0.0.1 /event/billing/adjustment/event 222875404996721006 0
Adjusted From:  0.0.0.1 /event/billing/product/fee/cycle/cycle_forward_monthly
222875404996719270 0
Service Type:   /service/ip
Event Time:     9/7/2025 20:6:54
----------
Resource          Amount         Disct     GL_ID
US Dollar         -180.000000    0.000     102
-----------------------------------------
Account: 0.0.0.1 /account 14854 0
Resource Name     Original Amount      New Amount      Difference
US Dollar         0.000000             -360.000000     -360.000000
Total Resources   0.000000             -360.000000     -360.000000
-------------------------------

++++++++++++++++++++++++++++++++++++
The Total Rerate Impact:
++++++++++++++++++++++++++++++++++++
Resource Name     Original Amount      New Amount      Difference
US Dollar         0.000000             -360.000000     -360.000000
Total Resources   0.000000             -360.000000     -360.000000
-----------------------------------------------
END OF REPORT
```

# Improving pin_rerate Performance

The following parameters can be used with **pin_rerate** to improve performance:

```
- pin_rerate  rerate_children  5
- pin_rerate  rerate_per_step  1000
- pin_rerate  rerate_fetch_size  5000
```

# 9
# Rerating Utilities

Learn about the syntax and parameters for the Oracle Communications Billing and Revenue Management (BRM) rerating utilities.

Topics in this document:



## load_pin_config_ood_criteria

Use the **load_pin_config_ood_criteria** utility to load out-of-order criteria into the **/config/event_order_criteria** object in the BRM database. You define out-of-order criteria in the **pin_config_ood_criteria.xml** file in *BRM_home***/sys/data/config**.

> **Note:**
>
> - The **load_pin_config_ood_criteria** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility.
> - The **load_pin_config_ood_criteria** utility overwrites existing out-of-order criteria. If you update out-of-order criteria, you cannot load new ones only. You must load complete sets of out-of-order criteria each time you run the **load_pin_config_ood_criteria** utility.

### Location

*BRM_home***/bin**

### Syntax

```
load_pin_config_ood_criteria [[-d][-v]|[-t] out_of_order_criteria_file.xml]|[-h]
```

### Parameters

**-h**
Displays online help about the command.

**-d**
Creates a log file for debugging purposes. Use this parameter when the utility appears to have run with no errors, but the data has not been loaded into the database.

**-v**
Displays detailed information as the utility runs.

**-t**
Checks the validity of the XML file but does not process any data.

***out_of_order_criteria_file***
The name and location of the file that defines out-of-order criteria. The default **pin_config_ood_criteria.xml** file is in *BRM_home***/sys/data/config**.
If you do not run the utility from the directory in which the file is located, you must include the complete path to the file. For example:

```
load_pin_config_ood_criteria BRM_home/sys/data/config/pin_config_ood_criteria.xml
```

> **Note:**
>
> If you copy the **pin_config_ood_criteria.xml** file to the directory from which you run the **load_pin_config_ood_criteria** utility, you do not have to specify the path or file name. By default, the name of the file is **pin_config_ood_criteria.xml**. You can change this name.

**Results**

The **load_pin_config_ood_criteria** utility notifies you when it successfully creates the **/config/event_order_criteria** object. Otherwise, look in the **default.pinlog** file for errors. This file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.

To verify that out-of-order criteria was loaded, you can display the **/config/event_order_criteria** object using the Object Browser application in Developer Center or the **robj** command with the **testnap** utility.

> **Note:**
>
> You must send a **Reload** semaphore to make new out-of-order criteria available in the Pipeline Manager. The **Reload** semaphore is used by the DAT_PortalConfig data module.

# load_pin_rerate_flds

Use this utility to load the following information into the BRM database:

- Load extraction keys-to-event field mappings into the **/config/rerate_flds** object. You define extraction keys and map them to EDR fields by using XML.
- Load balance-impact comparison fields into the **/config/rerate_flds/compare_bi** object. The balance impact fields in this object determine whether the balance impacts of rerating and previous rating are equivalent.

### Location

*BRM_home*/sys/data/config

Run **load_pin_rerate_flds** from this directory.

### Syntax

```
load_pin_rerate_flds -f xml_file_name [-v] [-h]
```

### Parameters

**-f *xml_file_name***
Specifies the name of the XML file that contains either the extraction keys and field mappings or the balance-impact comparison fields.

> **Note:**
>
> The file you load cannot include both extraction-key mapping and balance-impact comparison fields. You must load the files for these configurations separately.

**-verbose**
Displays information about successful or failed processing as the utility runs.

> **Note:**
>
> Always use this parameter with other parameters and commands. It is not position-dependent. For example, you can enter **-verbose** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the parameter. Replace *filename*.**log** with the name of the log file:
> **load_pin_rerate_flds** *any_other_parameter* **–v >** *filename*.**log**

**-help**
Displays the syntax and parameters for this utility.

### Results

This utility notifies you when it successfully creates the **/config/rerate_flds** object.

If it cannot create the **/config/rerate_flds**, it logs an error in the log file (**default.pinlog**). It creates the log file either in the directory from which the utility was started or in the directory specified in the configuration file.

# pin_event_extract

Use this utility to extract events from your database to be rerated by the BRM Pipeline Manager. See "Using Event Extraction Manager" for more information.

You define which events to extract by using the **pin_event_extract.cfg** file located in the *BRM_home*/apps/pin_event_extract directory.

> **Note:**
>
> The **pin_event_extract** utility needs a configuration file in the directory from which you run the utility.

### Location

*BRM_home*/**bin**

### Syntax

```
pin_event_extract [-f ConfigFileName] [-s] [-r roaming_input_file] [-b BrandName] [-e] [-o] [-h]
```

### Parameters

**-f *ConfigFileName***
Specifies the name and location of the configuration file that defines which events to extract. For information on how to create the file, look at the sample file (*BRM_home*/**apps/pin_event_extract/pin_event_extract.cfg**).
By default, the utility looks in the current working directory. If your configuration file is located in a different directory, you must specify the entire path for the file. If the **pin_event_extract.cfg** file is located in the current working directory, you can ignore this parameter.

**-s**
Extracts only events that match the event start time (**event_start_time_stamp**) specified in *ConfigFileName*.
You must use this option together with the **-f** parameter.

**-r *roaming_input_file***
Extracts events based on the information provided in *roaming_input_file*, which is generated by the RAP Processing pipeline during Outcollect processing. It contains records that specify the extraction criteria for extracting TAP settlement records from the BRM database.

**-b *BrandName***
Extracts events assigned to a specific brand. You can use this option alone or with the **-f** option.

**-e**
Flags the events for back-out-only rerating. Pipeline Manager backs out the balance impact and does not rerate the event.

**-o TRUE | FALSE**
Overrides the program lock.

- **-o TRUE** resets the status table to the unused state.
- **-o FALSE** sets the status table to the used state.

The Event Extraction Manager cannot run at the same time as Rated Event Loader or Rerated Event Loader. To prevent this, all three applications use a status table to determine if one of the other applications is running. If one of the applications terminates abnormally and leaves the status table in a locked state, the Event Extraction Manager cannot be started. In this case, use the **-o TRUE** option to reset the status table to an unused state.

> **Note:**
>
> Before using this option, ensure that the Rated Event Loader and Rerated Event Loader are stopped.

**-h**
Displays the syntax and parameters for this utility.

### Results

The **pin_event_extract** utility notifies you when it successfully completes a command.

# pin_load_rerate_jobs

This utility is used by the **OODHandler** batch handler to process out-of-order events. It converts rerate-request files generated by the FCT_EventOrder module into flist format and then calls PCM_OP_ACT_USAGE to generate a notification event.

> **Note:**
>
> You must run the utility from a directory that contains both the **pin_rerate_job_info.xsd** file and a **pin.conf** configuration file. By default, these files are located in the *BRM_home***/apps/pin_ood_handler/process** directory.

### Location

*BRM_home***/bin**

### Syntax

```
pin_load_rerate_jobs [[-d][-v]|[-t] rerate_request_file.xml]|[-h]
```

### Parameters

**-d**
Creates a log file for debugging purposes. Use this parameter for debugging when the utility appears to have run with no errors but the data has not been loaded into the database.

**-v**
Displays detailed information as the utility runs.

**-t**
Checks the validity of the XML file but does not process any data. When you run the file in test mode, it returns either that the XML file is valid or an error message that contains the invalid line number.

***rerate_request_file*.xml**
Specifies the name of the rerate request XML file that is generated by the FCT_EventOrder pipeline module.

**-h**
Displays online help about the command.

### Results

The utility notifies you only if it encounters errors. Look in the **default.pinlog** file for errors. This file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.

# pin_rate_change

Use this BRM utility after you change the rates for cycle forward and cycle forward arrears events in the current cycle. This utility triggers the creation of rerating requests that the **pin_rerate** utility uses to recalculate the charges for these events and adjust the balance impacts.

The **pin.conf** file for this utility is in *BRM_home***/apps/pin_rate_change**.

### Location

*BRM_home***/bin**

### Syntax

```
pin_rate_change [-v] [-d] [-h]
```

### Parameters

**-v**
Displays information about successful or failed processing as the utility runs.

> **Note:**
>
> This parameter is not position dependent. For example, you can enter **-v** at the beginning or end of a command to initiate the verbose parameter. To redirect the output to a log file, use the following syntax with the verbose parameter. Replace *filename***.log** with the name of the log file:
> **pin_rate_change** any_other_parameter **–v >** filename**.log**

**-d**
Creates a log file for debugging purposes. Use this parameter for debugging when the utility appears to have run with no errors, but the data has not been loaded into the database.

**-h**
Displays the syntax and parameters for this utility.

### Results

The **pin_rate_change** utility notifies you when it successfully creates the **rerating requests**.

If the **pin_rate_change** 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_rel

Loads batches of prerated or rerated event records from Pipeline Manager into the BRM database.

There are two ways to use this utility. When you initially run **pin_rel**, use the command without any options and use the file name as the only command-line parameter. You use the **override** option when the utility has not successfully completed its process and needs to be rerun.

**pin_rel** records any errors in the **pin_rel** log file (*BRM_home***/apps/pin_rel/rel.pinlog***)*.

### Location

*BRM_home***/apps/pin_rel**

### Syntax

```
pin_rel [-override] event_file_name
```

### Parameters

**-override**
This option starts a new **pin_rel** process. Use this option to restart **pin_rel** when it has abnormally stopped or you have stopped it manually.

> **Note:**
>
> Use this option only when you know there are no other RE Loader processes running.

RE Loader maintains a status of its operations. Because only one RE Loader process can load events into the same database at the same time, the status must indicate the loading process is complete before another process can start loading. If you manually stop **pin_rel**, its status may not reflect its true state. The **-override** parameter overrides the status and permits a new loading process to start, providing one is not already running.

***event_file_name***
The name of the event file to load, including its extension.

### Results

If **pin_rel** is successful, it returns PROCESSED SUCCESSFULLY in the RE Loader log file (*BRM_home***/apps/pin_rel/rel.pinlog**).

If an error occurs during loading, this utility cancels the loading process. An error is logged in the **rel.pinlog** file, SQL loader errors are logged in a “bad" file (*BRM_home***/apps/pin_rel/***CDR_file_name***.bad**), and the records loaded in this session are deleted from the database.

If an error occurs during account updating, the error is logged in the **rel.pinlog** file. Loaded records are not deleted from the database.

# pin_rerate

Use this BRM utility to rerate events rated in real-time and events rated in batch by Pipeline Manager, and update account balances with the rerated results.

If you rerate both real-time-rated and pipeline-batch-rated events concurrently, you run **pin_rerate** multiple times: once to create rerating jobs by selecting the accounts and events for rerating, and several times again to process the rerate jobs.

If you do not use Pipeline Manager for batch rating, you select the accounts and events and rerate them by using a single **pin_rerate** command.

> **Note:**
>
> - **pin_rerate** is a multithreaded application.
> - In a multischema environment, you must run **pin_rerate** separately on each database schema.
> - To connect to the BRM database, the **pin_rerate** utility needs a configuration file in the directory from which you run the utility.

### Location

*BRM_home*/**bin**

The **pin.conf** file for this utility is located in *BRM_home*/**apps/pin_rerate**. Run **pin_rerate** from this directory.

### Syntax

```
pin_rerate      [-t [ss/mm/hh/] MM/DD/YYYY]
                [-a account POID_id]
                [[-d | -g | -i | -m | -n | -p | -s] input_file]
                [-line subscription_id]
                [-field_name input_file]
                [-r]
                [-reason reason_code]
                [-b [c | e]]
                [-c]
                [-backout]
                [-e  [-a | -s | -p | -n | -d]]
                [-h | help]
                [-purge [-t [ss/mm/hh/] MM/DD/YYYY]]
                [-l [d | s | sd | n]]
                -process jobs [-reason reason_code [,reason_code...]]
                -process queue
                -rerate [-reason reason_code [,reason_code...][-l [d | s | sd | n]]
                -process recycle [-db database_id]
```

### Parameter Overview

The parameters specify which accounts and events to rerate and which rerating process to perform.

- To select the accounts and events for rerating, see the following sections:

  - Parameters for Selecting Accounts for Rerating
  - Parameter for Specifying Events for Rerating
- To assign a rerate reason code for rerating, see "Parameter for Assigning a Rerate Reason".
- To specify rerating behavior, such as how to order the events, whether to update the database, and whether to generate reports, see "Parameters Affecting Rerating Behavior".
- To purge rerate jobs, see "Parameter for Purging Rerate Jobs".
- To process rerate jobs and rerate the selected accounts and events, see "Parameters for Processing Rerate Jobs".

### Parameters for Selecting Accounts for Rerating

The following parameters are used to select accounts for rerating based on the described event criteria.

All account selection parameters are optional and mutually exclusive, except for the **-t** parameter, which is mandatory when selecting the accounts and events to rerate. If you specify only the time (**-t**) parameter, BRM selects all accounts for rerating in the period defined by the **-t** parameter.

**-a [account *POID_id*]**
Selects a *single account* for rerating based on the provided account POID.
When you specify an account POID, you can also use the **-c** option to rerate events *without* updating the database.
This parameter is optional.

> **Note:**
>
> You cannot use this parameter with the **-d, -g, -i, -m, -p, -s**, **-line**, or custom parameters.

**-d *input_file***
Selects accounts for rerating based on the *bundle*. Accounts that have events associated with the charge offers and discount offers that belong to the bundles specified in *input_file* are selected for rerating.
*input_file* format: A text file with one bundle name specified on each line. The bundle names are case sensitive.
This parameter is optional.

> **Note:**
>
> - If a charge offer is part of multiple bundles, all accounts that purchase the charge offer are selected for rerating even if they did not purchase the bundle that was rerated.
> - You cannot use this parameter with the **-a, -g, -i, -m, -n, -p, -s**, **-line**, or custom parameters.

**-g *input_file***
Selects accounts for rerating based on one of the following:

- The account and bill unit objects
- The account, bill unit, and balance group objects

Accounts that match any criteria listed in *input_file* are selected for rerating.

- If items in *input_file* specify an account, bill unit, and balance group, only events specific to the balance group are selected.
- If items in *input_file* specify an account and a bill unit only, **pin_rerate** searches for all the balance groups associated with the bill unit and then selects the events specific to those balance groups.

*input_file* format: A text file containing information for one or more accounts in flist format. Specify an account POID, **/billinfo** POID, and balance group POID in a results array for each account. For example:

```
0 PIN_FLD_RESULTS                  ARRAY [1]
    1   PIN_FLD_ACCOUNT_OBJ         POID [0] $DB_NO /account 12345 0
    1   PIN_FLD_BILLINFO_OBJ        POID [0] $DB_NO /billinfo 34567 0
    1   PIN_FLD_BAL_GRP_OBJ         POID [0] $DB_NO /balance_group 66765 0
0 PIN_FLD_RESULTS                  ARRAY [1]
    1   PIN_FLD_ACCOUNT_OBJ         POID [0] $DB_NO /account 12344 0
    1   PIN_FLD_BILLINFO_OBJ        POID [0] $DB_NO /billinfo 45654 0
    1   PIN_FLD_BAL_GRP_OBJ         POID [0] $DB_NO /balance_group NULL 0
...
```

where:

- PIN_FLD_ACCOUNT_OBJ specifies the POID of the account object.
- PIN_FLD_BILLINFO_OBJ specifies the POID of the bill unit object.
- PIN_FLD_BAL_GRP_OBJ specifies the POID of the balance group object.

To rerate all events for all balance groups for a given account's bill unit, specify a value of NULL as the POID ID for the balance group object, as shown in the second results array in the above example.

This parameter is optional.

> **Note:**
>
> You cannot use this parameter with the **-a**, **-d**, **-i**, **-m**, **-n**, **-p**, **-s**, **-e**, **-line**, or custom parameters.

**-i *input_file***
Selects accounts for rerating based on the *discount object*. Accounts that own at least one instance of the discount objects specified in *input_file* are selected for rerating.
*input_file* format: A text file with one discount name specified on each line. Discount names are case sensitive.
This parameter is optional.

> **Note:**
>
> You cannot use this parameter with the **-a**, **-d**, **-g**, **-m**, **-n**, **-p**, **-s**, **-line**, or custom parameters.

**-m *input_file***
Selects a set of accounts for rerating based on the provided account POIDs.
*input_file* format: A text file containing the accounts to rerate in flist format. Specify each account in a results array. For example:

```
0 PIN_FLD_RESULTS                   ARRAY [1]
    1 PIN_FLD_ACCOUNT_OBJ            POID [0] $DB_NO /account 12345 0
0 PIN_FLD_RESULTS                   ARRAY [2]
    1 PIN_FLD_ACCOUNT_OBJ            POID [0] $DB_NO /account 12333 0
...
```

where PIN_FLD_ACCOUNT_OBJ is the POID of the account object.

> **Note:**
>
> You cannot use this parameter with the **-a**, **-d**, **-g**, **-i**, **-n**, **-p**, or **-s -e**, **-line**, or custom parameters.

**-n *input_file***
Selects accounts for rerating based on the *event types*. Accounts that have events of the types specified in *input_file* are selected for rerating.
*input_file* format: A text file with one event type specified on each line. Event types are case sensitive.
This parameter is optional.

> **Note:**
>
> - You cannot use this parameter with the **-a**, **-d**, **-g**, **-i**, **-m**, **-p**, **-s**, **-line**, or custom parameters.
> - If you use this parameter with the **-r** parameter, all subclasses of the event type specified in the input file are also rerated.
> - When using a custom **pin_rerate** parameter, the **-n** parameter is mandatory if the custom parameter is based on an event field that is present only in an event subclass. In this case, the **-n** parameter input file can contain only one event type.

**-p *input_file***
Selects accounts for rerating based on the *charge offer*. Accounts that have events associated with the charge offers specified in *input_file* are selected for rerating.
*input_file* format: A text file with one charge offer name specified on each line. Charge Offer names are case sensitive.
This parameter is optional.

> **Note:**
>
> You cannot use this parameter with the **-a**, **-d**, **-g**, **-i**, **-m**, **-n**, **-s**, **-line**, or custom parameters.

**-s *input_file***
Selects accounts for rerating based on the *service type*. Accounts that have events associated with the service types specified in *input_file* are selected for rerating.
*input_file* format: A text file with one service type specified on each line. Service type names are case sensitive.
This parameter is optional.

> **Note:**
>
> You cannot use this parameter with the **-a**, **-d**, **-g**, **-i**, **-m**, **-n**, **-p**, **-line**, or custom parameters.

**-t [ss/mm/hh/]MM/DD/YYYY**
Selects accounts for rerating based on the event *start time*. All accounts with events that occurred between the start time and the current date are selected for rerating.

> **Note:**
>
> The group of selected accounts can be further restricted by using criteria specified with the **-a**, **-d**, **-g**, **-i**, **-m**, **-n**, **-p**, **-s**, **-line**, or custom parameters.

**-line *subscription_id***
Selects accounts for rerating based on a subscription service.
*subscription_id* specifies the subscription service ID. The subscription service is identified by the PIN_FLD_ALIAS_LIST.PIN_FLD_NAME field, which can specify, for example, the caller ID such as a phone number or the MAC address.
This parameter is optional.

> **Note:**
>
> The group of selected accounts can be further restricted by using criteria specified with the **-a**, **-d**, **-g**, **-i**, **-m**, **-n**, **-p**, **-s**, or custom parameters.

***-field_name input_file***
*field_name* is a custom **pin_rerate** parameter you have defined that maps to a specific event field.
*input_file* is a file that contains the values of the event field that are used to select the accounts for rerating.
Selects accounts based on the custom parameter. Accounts are selected for rerating that have events with the field identified by *field_name* whose field value is one of those specified in *input_file*.

> **Note:**
>
> - You cannot use this parameter with the **-a**, **-d**, **-g**, **-i**, **-m**, **-p**, **-s**, or **-line** parameters.
> - If the custom parameter maps to a field in an **/event** subclass, you must specify the subclass event type by including the **-n** parameter, and the **-n** parameter input file can include only one event type. If you specify more than one event type, an error is returned.

### Parameter for Specifying Events for Rerating

Use the following parameter to specify which events to rerate for the selected accounts.

**-r**
Specifies whether to use *selective rerating*. This applies the account selection criteria parameters (**-a**, **-d**, **-g**, **-i**, **-m**, **-n**, **-p**, **-s**, **-line** or *field_name*) to the account's events as well and selects events for rerating based on the specified parameter. For example, using **-r** and **-p** *input_file* in the command line selects accounts based on the charge offers specified in *input_file*, and then selects and rerates only the account's events associated with those charge offers.

> **Note:**
>
> Do not use selective rerating if your rating scheme includes credit limits or balance element-based tiers. These schemes require that all events related to an account are rerated to ensure accurate rerating.
> Do not use selective rerating if deferred taxation is used for taxing events during rerating.
> Use caution when specifying selective rerating. Account balances can be impacted by different types of events, and the order of the balance impacts is important to accurately apply discounts and consume balance elements. It is typically safer to rerate all of an account's events.

This parameter is optional and can be used with any other parameter. The value for this parameter can be specified as either **0 (disable)** or **1 (enable)**.

> **Note:**
>
> If no value is specified for this parameter, the default behavior is to consider the value as **1**.

### Parameter for Assigning a Rerate Reason

**-reason *reason_code***
Use with other selection parameters to assign a rerate reason code to the jobs created for the selected accounts.
Reason code format: Any integer except **1** (**1** is a reserved default value for pipeline-generated rerating requests).

If you create rerate jobs through BRM automatic rerating, rerate reasons are hard coded by the opcodes that handle those rerating scenarios. Be sure your rerate reasons are unique to process rerate jobs associated with them during separate rerating process.
To process rerate jobs according to a rerate reason, see "Parameters for Processing Rerate Jobs".

### Parameters Affecting Rerating Behavior

The following are additional, optional parameters that affect rerating behavior and the utility's output.

**-b [ c | e ]**
Specifies the order in which *batch* events are rated. The **c** option rerates events in the order that they were created in the BRM database. The **e** option rerates events based on the time when the event actually occurred. The default is **e**.

> **Note:**
>
> Due to real time account balance impacts and credit limit monitoring, the order in which the batch events are processed may result in different rating results.

This parameter is optional.

**-c**
Specifies calculation of rerating only. Use this parameter to log the rerating result of an account into a report file without updating the database. This option can be used for only one account, so it works only when using the **-a** parameter.
This parameter is optional.

**-backout**
Specifies back-out-only rerating, which backs out the balance impacts of rating without rerating events.
This parameter is optional and can be used with any other parameter.

> **Note:**
>
> When choosing the events to back out, ensure that you do not select events that could imbalance your general ledger, such as events for which fees have already been paid and usage events that should be rerated. Typically, back-out-only rerating is performed only on usage events where rating should not have occurred.

**-e [-a | -s | -p | -n | -d ]**
Returns an estimate of how many events might be affected by rerating based on which accounts are being rerated. Options:
**-e -a**: A single account
**-e -s**: Accounts with a specific service type
**-e -p**: Accounts with specific charge offers
**-e -n**: Accounts with specific event types
**-e -d**: Accounts with specific bundles

> **Note:**
>
> If you do not specify one of these options when using the **-e** parameter, the utility returns **0**.

The **pin_rerate** utility prints the output of this option on the screen or to the **pin_rerate.pinlog** file in **$PIN_LOG/pin_rerate**.
This parameter is optional.

> **Note:**
>
> You cannot use this parameter with the **-m** or **-g** parameters.

**-h | help**
Displays the syntax and parameters for this utility.

**-l [ d | s | sd | n ]**
Specifies whether to generate a report. Options:
**-ld**: Specifies a detailed report (**pin_rerate.detail**).
**-ls**: Specifies a summary report (**pin_rerate.summary**).
**-lsd**: Specifies both detailed and summary reports. This is the default.
**-ln**: Specifies no report.

> **Note:**
>
> When Pipeline Manager is running, you can specify these report options at rerate time (when you use the **-rerate** parameter). You cannot specify them when you create the rerate jobs.

#### Parameters for Processing Rerate Jobs

You use different parameters to process rerate jobs depending on whether you rerate real-time-rated and pipeline-batch-rated events concurrently or, if you do not use Pipeline Manager for batch rating, rerate only real-time-rated events.

- Processing Rerate Jobs When Rerating Real-Time and Pipeline Rated Events Concurrently
- Processing Rerate Jobs When You Do Not Use Pipeline Manager

#### Processing Rerate Jobs When Rerating Real-Time and Pipeline Rated Events Concurrently

To process rerate jobs when rerating real-time and pipeline rated events concurrently, you run the following commands in the order shown.

```
pin_rerate -process jobs [-reason comma_separated_reason_codes]
pin_rerate -process queue
pin_rerate -rerate [-l  [d | s | sd | n] ]
pin_rerate -process queue
pin_rerate -process recycle
```

> **Note:**
>
> You must run **pin_rerate** with all of the above parameters to complete the rerating process.

These commands process rerate jobs that were created both manually, by running **pin_rerate** to select accounts for rerating (see "Parameters for Selecting Accounts for Rerating"), and automatically, by BRM automatic rerating features.

The parameters specify the following:

**-process jobs [-reason *comma_separated_reason_codes*]**
Searches for all rerate jobs with status equal to NEW, and notifies Pipeline Manager to lock the accounts in the rerate jobs. Then updates the status of the rerate jobs to WAITING_ACCOUNT_LOCK.
If you use the **-reason** parameter, performs the same actions but only for the rerate jobs associated with the specified reason codes, where:

***comma_separated_reason_codes***
Specifies the codes you assigned as the rerate reason for the jobs you want to process.

> **Note:**
>
> If you do not specify a rerate reason code, all rerate jobs in the rerate queue are processed.

**-process queue**
Processes acknowledgment events from Pipeline Manager and updates rerate job status as follows:

- If acknowledging notification to suspend EDRs, it updates the status to ACCOUNT_LOCKED.
- If acknowledging notification to resume processing EDRs, it updates the status to READY_TO_RECYCLE.

This parameter is used twice: before and after the **-rerate** parameter.

**-rerate [-l  [d | s | sd | n]]**
Rerates events for the accounts referenced in the rerate jobs, and updates the rerate job status to RERATED.

**-l**
Specifies to generate a rerating report. Options are:
**-ld**: Generates a detailed report (**pin_rerate.detail**).
**-ls**: Generates a summary report (**pin_rerate.summary**).
**-lsd**: Generates both detailed and summary reports. This is the default.
**-ln**: Generates no report.

**-process recycle [-db *database_id*]**
Recycles the EDRs suspended during the rerating process, and updates the rerate job status to COMPLETE.

-**db** *database_id*: Specifies the ID of the database schema that contains the suspended EDRs in the format *x.x.x.x* (for example, 0.0.0.2). Use this only if the suspended EDRs are not stored in the current BRM schema.

> **Note:**
>
> If you do not specify the **-db** *database_id* parameter, the suspended EDRs are picked up from the current schema.

#### Processing Rerate Jobs When You Do Not Use Pipeline Manager

If you do not use Pipeline Manager for batch rating, rerate jobs for real-time-rated events are processed as follows:

- When you run **pin_rerate** with parameters that specify the accounts and events to rerate, **pin_rerate** creates the rerate jobs for those accounts are then immediately rerates their events.
- Rerate jobs that already exist (such as those created by automatic rerating) are processed with the following parameter:

**-rerate [-reason *comma_separated_reason_codes*]**
Rerates events for the accounts referenced in the rerate jobs, and updates the rerate job status to COMPLETE.
If you use the **-reason** parameter, performs the same actions but only for the rerate jobs associated with the specified reason codes.
*comma_separated_reason_codes*
Specifies the codes you assigned as the rerate reason for the jobs you want to process.

> **Note:**
>
> If you do not specify a rerate reason code, all rerate jobs in the rerate queue are processed.

#### Parameter for Purging Rerate Jobs

Use the following parameter to purge rerate jobs.

**-purge [-t [*ss/mm/hh/*] *MM/DD/YYYY*]**
Purges rerate jobs that have a rerate job status of COMPLETE or UNSUCCESSFUL.
[**-t** [*ss/mm/hh/*] *MM/DD/YYYY*]
Specifies to purge rerate jobs that have had their rerate job status set to COMPLETE or UNSUCCESSFUL before the time specified.
If no time is specified, all rerate jobs that have a rerate job status of COMPLETE or UNSUCCESSFUL are purged up to the current date.

#### Results

If the **pin_rerate** utility does not notify you that it was successful, look in the utility log file (**pin_rerate.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.

If rerating fails, the utility creates a report that includes the account numbers and start times for failed rerates. The report file name is **pin_rerate.status_report**, and is in the directory from where you ran the utility.