# Oracle® Communications Billing and Revenue Management

## Collecting Revenue Assurance Data



Release 16.26.0
G54449-01
October 2026



Oracle Communications Billing and Revenue Management Collecting Revenue Assurance Data, Release 16.26.0

G54449-01

Copyright © 2017, 2026, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

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.

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.

# Contents


About This Content

1 Understanding Revenue Assurance Manager

About Revenue Assurance Manager 1
About Collecting Revenue Assurance Data 1
How Revenue Assurance Data Is Collected 2
About Viewing Revenue Assurance Data 3
About Splitting Revenue Assurance Data into Detailed Categories 3
About Split Revenue Assurance Data 4
Splitting Revenue Assurance Data by Type of Charge 5
About Collecting Revenue Assurance Data on Write-Offs 5
About Using Event Notification for Revenue Assurance 5

2 Installing Revenue Assurance Manager

Installing Revenue Assurance Manager 1
Installing Revenue Assurance Center 2
Starting and Using Revenue Assurance Center 2
Customizing Revenue Assurance Center 2
Uninstalling Revenue Assurance Manager 3

3 Generating Revenue Assurance Data for Billing

Configuring BRM to Generate Revenue Assurance Data for Billing 1
Enabling Billing Utilities to Generate Revenue Assurance Data 1
Configuring Custom Non-MTA Applications to Generate Revenue Assurance Data 2
Enabling Billing Operations to Generate Revenue Assurance Data 3
Configuring Item Subtypes to Store Split Revenue Assurance Data 4

4 Collecting Revenue Assurance Data for Usage Write-Offs

Configuring BRM to Collect Revenue Assurance Data for Usage Write-Offs 1
Configuring Event Notification for Revenue Assurance 1




Copyright © 2017, 2026, Oracle and/or its affiliates.

Renaming the Control Point for Collecting Data from Usage Write-Offs 2

5 Using Batch IDs in Revenue Assurance Manager

Tracking Event Records by Batch IDs 1
Keeping Track of Rejected Event Records by Batch IDs 2

6 Configuring Revenue Assurance Alerts

About Revenue Assurance Alerts 1
Configuring Alert Thresholds 1
Specifying Threshold Data and Time Periods 2
Specifying Threshold Filter Conditions 3
Specifying Threshold Field Value Ranges 3
Alert Threshold Examples 4
Period Alert Example 4
Daily Alert Example 4
Weekly Alert Example 4
Loading Alert Details into the Database 5
Loading Alert Details by Using load_pin_config_ra_thresholds 5
Configuring Email Alerts 5
Specifying Email Alert Text 6
Specifying Email Alert Details 7
Generating Revenue Assurance Alerts 8

7 Revenue Assurance Utilities

load_ara_config_object 1
load_pin_config_ra_alerts 2
load_pin_config_ra_flows 3
load_pin_config_ra_thresholds 4
pin_ra_check_thresholds 5


Copyright © 2017, 2026, Oracle and/or its affiliates.

# About This Content

This document describes how to use Oracle Communications Billing and Revenue Management (BRM) Revenue Assurance Manager and Revenue Assurance Center to prevent revenue leakage in BRM billing and write-off processes.

### Audience

This guide is intended for operations personnel and system administrators.


Copyright © 2017, 2026, Oracle and/or its affiliates.

1

# Understanding Revenue Assurance Manager

Learn how to collect revenue assurance data in Oracle Communications Billing and Revenue Management (BRM) by using Revenue Assurance Manager.

Topics in this document:


- About Revenue Assurance Manager
- About Collecting Revenue Assurance Data
- About Collecting Revenue Assurance Data on Write-Offs
- About Using Event Notification for Revenue Assurance


> **Note**
>
> Revenue Assurance Manager is an optional feature that you install separately. See "Installing Revenue Assurance Manager".

## About Revenue Assurance Manager

You use Revenue Assurance Manager to do the following:

- **Verify Completeness and Accuracy**: Ensure the completeness, accuracy, and integrity of revenue-related processes in BRM. See "About Collecting Revenue Assurance Data".
- **Collect Data on Write-Offs**: Gather revenue assurance data pertaining to write-offs. See "About Collecting Revenue Assurance Data on Write-Offs".
- **Set Up Alerts**: Configure alerts to notify you when revenue assurance data exceeds specified thresholds. For example, you can set an alert to trigger when the number of suspended event records exceeds 100,000. See "About Revenue Assurance Alerts".

The data collected by Revenue Assurance Manager is stored in the BRM database. You can analyze this data to identify revenue leakage within your system. You can view the data by using Revenue Assurance Center or Revenue Assurance Manager reports. See "About Viewing Revenue Assurance Data".

## About Collecting Revenue Assurance Data

You can collect revenue assurance data from the following BRM processes:

- Regular billing:
  - Billing accounts (**pin_bill_accts** utility)
  - Collecting credit and debit card payments (**pin_collect** utility)
  - Invoicing (**pin_inv_accts** utility)
  - Deferring actions (**pin_deferred_act** utility)



Copyright © 2017, 2026, Oracle and/or its affiliates.

  - Calculating cycle fees (**pin_cycle_fees** utility)
- Bill on purchase
- Bill now
- Auto-triggered billing
- Trial billing (**pin_trial_bill_accts** utility)

To collect revenue assurance data from those processes, you must configure the related utilities and operations to generate that data. See "Configuring BRM to Generate Revenue Assurance Data for Billing".

Revenue assurance data consists of statistics such as the following:

- Number of input records
- Number of successful records
- Number of failed records
- Total revenue billed
- Total revenue to be collected
- Total revenue collected
- Number of successfully billed accounts
- Number of accounts that failed to be billed
- The total revenue associated with successfully billed accounts
- The number of suppressed bills
- The revenue associated with suppressed bills

BRM stores revenue assurance data in **/process_audit** and **/process_audit/billing** objects in the BRM database.

After revenue assurance data has been collected, you can view the data by using Revenue Assurance Center or Revenue Assurance reports. See "About Viewing Revenue Assurance Data".

# How Revenue Assurance Data Is Collected

Revenue Assurance Manager collects data from the following:

- Bill on purchase, bill now, and auto-triggered billing operations.
- Control points in utilities. The control points are preconfigured, and you cannot change them.

  Table 1-1 lists the utility control points.

**Table 1-1 Utility Control Points**

| Control Point | Utility |
|---|---|
| Revenue Anticipated | The **pin_bill_accts** utility that selects accounts ready for billing provides information on anticipated revenue. |
| Revenue Collected | The **pin_collect** utility that collects the balance due for bills paid by credit card or direct debit provides information on collected revenue. |


Copyright © 2017, 2026, Oracle and/or its affiliates.

**Table 1-1 (Cont.) Utility Control Points**

| Control Point | Utility |
|---|---|
| Invoice Generated | The **pin_inv_accts** utility that creates a bill for the balance due provides information on the number of generated invoices, including total event records processed, event records successfully invoiced, and failed event records. |
| Scheduled Deferrals Executed | The **pin_deferred_act** utility that processes the deferred actions calculates the number of processed scheduled deferrals. |
| Cycle Fees Generated | The **pin_cycle_fees** utility used to identify cycle forward fees that have reached the end of free billing periods calculates the number of generated cycle fees. |

# About Viewing Revenue Assurance Data

You can view revenue assurance data by using either of the following:

- **Revenue Assurance Manager reports:** You can generate summary and detail reports from revenue assurance data. You can change the report parameters to meet your business needs.

  See "Revenue Assurance Manager Reports" in *BRM Reports*.

- **Revenue Assurance Center:** You can use this application in Business Operations Center to search for and view revenue assurance statistics. See Revenue Assurance Center Help for detailed instructions about using the application.

  See "Installing Revenue Assurance Center".

When you use Revenue Assurance Center to view revenue assurance data, you can search for data based on the following information:

- Billing segments
- Accounts billed on the same accounting day of the month (DOM)
- A combination of billing segment and DOM

You can display data from all or some of the control points listed in Table 1-1.

> **Note**
>
> The data that can be displayed in Revenue Assurance Center is restricted as follows:
>
> - Revenue assurance data collected from trial billing is viewable only in reports.
> - Revenue Assurance Center can display only split revenue assurance data. See "About Splitting Revenue Assurance Data into Detailed Categories".

# About Splitting Revenue Assurance Data into Detailed Categories

You can split revenue assurance data into detailed categories, such as amounts associated with A/R items, billable items, and suppressed bills. Splitting revenue assurance data enables you to pinpoint the source of revenue leakage in your system more accurately.

The following operations automatically generate split revenue assurance data:


Copyright © 2017, 2026, Oracle and/or its affiliates.

- Bill now
- Bill on purchase
- Auto-triggered billing

> **Note**
>
> By default, the generation of revenue assurance data by the preceding operations is disabled. To enable it, see "Enabling Billing Operations to Generate Revenue Assurance Data".

The following utilities have a **-split** parameter that you can use to separate revenue assurance data:

- **pin_bill_accts**
- **pin_trial_bill_accts**

> **Note**
>
> If you intend to collect revenue assurance data from these utilities, be sure to use the **-split** parameter when you run them.

See the discussions on **pin_bill_accts** and **pin_trial_bill_accts** in *BRM Configuring and Running Billing*.

You can view split revenue assurance data in Revenue Assurance Center or in Revenue Assurance reports:

- Revenue Assurance Center automatically displays split data. It cannot display data that is not split.
- You can configure Revenue Assurance Reports to display split or unsplit data. See "About the Split Method Parameter" in *BRM Reports*.

## About Split Revenue Assurance Data

Split revenue assurance data includes the following categories:

- **A/R items:** If service types are associated with these items, the amount for each service type is also generated:
  - **Adjustments:** Adjustment charges.
  - **Payments:** Total number of payments received.
  - **Payment reversals:** Total payment reversals.
  - **Disputes:** Sum of disputed amounts.
  - **Settlements:** Sum of settlements.
  - **Refunds:** Total refunds.
  - **Write-offs:** Total number of written-off accounts.
  - **Write-off reversals:** Total number of reversed account write-offs.


Copyright © 2017, 2026, Oracle and/or its affiliates.

- **Billable items:** The billable items and their subtypes that you configured (for example, usage charges, recurring charges, one-time charges, and discounts).
- **Miscellaneous charges:** If billable items are not configured, the split revenue assurance data is stored as miscellaneous charges.
- **Connections:** The number of billed subscription devices or services.
- **Bills suppressed:** The number of suppressed bills, the reason for the suppression, and the anticipated revenue.

> **Note**
>
> The bill-suppressed data is generated by regular billing, trial billing, and auto-triggered billing only.

### Splitting Revenue Assurance Data by Type of Charge

You can also split revenue assurance data by the type of charge, such as discounts and usage, one-time, and recurring charges. You do this by setting up custom billable items and item subtypes.

If you do not configure billable item subtypes, the split revenue assurance data is stored as miscellaneous charges.

See "Configuring Item Subtypes to Store Split Revenue Assurance Data".

## About Collecting Revenue Assurance Data on Write-Offs

You can use Revenue Assurance Manager to collect statistics about the following types of write-offs:

- Written-off accounts, bills, and bill items

  This information is gathered from preconfigured utility control points and is displayed as A/R split revenue assurance data. See "About Splitting Revenue Assurance Data into Detailed Categories".

- Written-off usage events

  When a usage event is written off through Suspense Manager, an **/event/notification/suspense/writeoff** event is generated. You configure event notification so that Revenue Assurance Manager collects data every time such an event occurs. See "About Using Event Notification for Revenue Assurance".

  The data collected from a written-off usage event includes the original batch ID and the number of event records that were written off in that batch of records.

  See "Configuring BRM to Collect Revenue Assurance Data for Usage Write-Offs".

## About Using Event Notification for Revenue Assurance

Revenue Assurance Manager uses event notifications to collect data on written-off usage events. The following events are specifically generated to support the revenue assurance event notification process:

- **/event/notification/suspense/writeoff** — This event is triggered when suspended usage event records are written off. When Suspense Manager generates this event the system


Copyright © 2017, 2026, Oracle and/or its affiliates.

calls the PCM_OP_PROCESS_AUDIT_CREATE_WRITEOFF_SUMMARY opcode by default.

- **/event/notification/ra_threshold** — This event is generated when specific conditions for generating revenue leakage alerts are met. The **load_pin_config_ra_thresholds** utility generates this event, which triggers a call to the PCM_OP_PROCESS_AUDIT_POL_ALERT policy opcode. See "load_pin_config_ra_thresholds".

See "Configuring Event Notification for Revenue Assurance".


Copyright © 2017, 2026, Oracle and/or its affiliates.

2

# Installing Revenue Assurance Manager

Learn how to install Oracle Communications Billing and Revenue Management (BRM) Revenue Assurance Manager and Business Operations Center (which includes Revenue Assurance Center).

Topics in this document:


- Installing Revenue Assurance Manager
- Installing Revenue Assurance Center
- Starting and Using Revenue Assurance Center
- Customizing Revenue Assurance Center
- Uninstalling Revenue Assurance Manager


> **Note**
>
> - Revenue Assurance Manager and Business Operations Center are optional features that you install separately.
> - Revenue Assurance Center is a component of Business Operations Center. When you install Business Operations Center, you install Revenue Assurance Center.

## Installing Revenue Assurance Manager

> **Note**
>
> If you already have Revenue Assurance Manager installed, you must uninstall it before installing a new version.

The Revenue Assurance Manager server component is supported on the Linux operating system. For information on disk space requirements for those operating systems, see *BRM Installation Guide*.

To install Revenue Assurance Manager, see "Installing Individual BRM Components" in *BRM Installation Guide*.

After installing Revenue Assurance Manager, if your event tables are partitioned, run the **partition_utils** utility with the **-o update** parameter from the *BRM_home*/**apps/partition_utils** directory:

```
perl BRM_home/bin/partition_utils -o update
```

See "Updating Partitions" and "partition_utils" in *BRM System Administrator's Guide*.



Copyright © 2017, 2026, Oracle and/or its affiliates.

# Installing Revenue Assurance Center

Revenue Assurance Center is supported on the Windows platform and requires approximately 100 MB of disk space.

Before installing Revenue Assurance Center, obtain information such as the following about the machine running the CM to which Revenue Assurance Center connects:

- Location for the software (if different from the default)
- Computer or server name
- Port number
- Database number

To install Revenue Assurance Center, see "Installing Individual BRM Clients" in *BRM Installation Guide*.

# Starting and Using Revenue Assurance Center

To start and use Revenue Assurance Center, see "Starting a BRM Client Application on Windows" in *BRM Installation Guide*.

See "Customizing Revenue Assurance Center" for information on how you can change features in Revenue Assurance Center.

# Customizing Revenue Assurance Center

You can use the **racustom.properties** file to customize some features in Revenue Assurance Center. The file is located in *BRM_Client_home***RevenueAssuranceCenter\lib**, where *BRM_Client_home* is the directory in which the BRM clients are installed.

The **racustom.properties** file contains the following text:

```
# racustom.properties
# Revenue Assurance Center's customizable properties file
# RA Details screen default view by option is configured using the # following key. The
valid value for this key should be any of 0 #or 1 or 2 or 3 or 4
#Service Type = 0
#BatchID      = 1
#Revenue Stream = 2
#Output Stream   = 3
#Suspense Reason = 4
ra.ratingdetail.default.viewby.value=0
ra.reratingdetail.default.viewby.value=0
ra.status.success=Success
ra.status.suspended=Suspense
ra.status.writtenoff=WRITTEN-OFF
ra.searchdlg.rating.servicetype.items=SMS,GSM
ra.searchdlg.rerating.servicetype.items=SMS,GSM
ra.common.currency.symbol=$
ra.common.decimal.separator=.
ra.common.currentlocale=en
```

The file specifies the following:

- The option settings from the Revenue Assurance Center Customize Table dialog box.


Copyright © 2017, 2026, Oracle and/or its affiliates.

- The default grouping parameter for the Revenue Assurance Center Details screen. The Details screen displays revenue assurance data organized by the default grouping parameter.

  You can manually change this default configuration by changing the number value in strings of this type:

  ```
  ra.ratingdetail.default.viewby.value=0
  ```

  The preceding line shows Service Type selected as the default. If you change **0** to **2**, your default option becomes **Revenue Stream**.

- The **Service Type** search options in the rating and rerating Search dialog box.

  Add service types to the following lines to add search options to the dialog box:

  ```
  ra.searchdlg.rating.servicetype.items=SMS,GSM
  ra.searchdlg.rerating.servicetype.items=SMS,GSM
  ```

- The default system currency. The default is US dollars (ISO code 840); you can change this to any other currency type. To change the default currency to euros, use the following unicode escape sequence code:

  ```
  ra.common.currency.symbol=\u20AC
  ```

- The default decimal separator (**.**) for the default currency. If the default currency is changed from US dollars to euros, change the decimal separator to a comma (**,**):

  ```
  ra.common.decimal.separator=,
  ```

- The current locale of the BRM server in which the CM is running. You cannot change this locale. The default setting is English:

  ```
  ra.common.currentlocale=en
  ```

# Uninstalling Revenue Assurance Manager

To uninstall Revenue Assurance Manager, see "Uninstalling Optional Components" in *BRM Installation Guide*.

To uninstall Revenue Assurance Center, see "Uninstalling BRM Client Applications" in *BRM Installation Guide*.


Copyright © 2017, 2026, Oracle and/or its affiliates.

3

# Generating Revenue Assurance Data for Billing

Learn how to configure Oracle Communications Billing and Revenue Management (BRM) to generate revenue assurance data for billing.

Topics in this document:

- Configuring BRM to Generate Revenue Assurance Data for Billing

## Configuring BRM to Generate Revenue Assurance Data for Billing

> **Note**
>
> To generate revenue assurance data, you must have Revenue Assurance Manager installed. See "Installing Revenue Assurance Manager".

Configuring BRM to generate revenue assurance data for billing consists of the following high-level tasks:

1. Enabling Billing Utilities to Generate Revenue Assurance Data
2. Configuring Custom Non-MTA Applications to Generate Revenue Assurance Data
3. Enabling Billing Operations to Generate Revenue Assurance Data
4. Configuring Item Subtypes to Store Split Revenue Assurance Data

### Enabling Billing Utilities to Generate Revenue Assurance Data

The utilities listed in Table 3-1 contain preconfigured control points that can generate revenue assurance data.

**Table 3-1 Utility Configuration Files**

| Utility | Configuration File |
|---|---|
| **pin_bill_accts** | *BRM_home*/apps/pin_billd/pin.conf |
| **pin_collect** | *BRM_home*/apps/pin_billd/pin.conf |
| **pin_inv_accts** | *BRM_home*/apps/pin_inv/pin.conf |
| **pin_deferred_act** | *BRM_home*/apps/pin_billd/pin.conf |
| **pin_cycle_fees** | *BRM_home*/apps/pin_billd/pin.conf |
| **pin_trial_bill_accts** | *BRM_home*/apps/pin_trial_bill/pin.conf |


Copyright © 2017, 2026, Oracle and/or its affiliates.

By default, however, those utilities do not generate revenue assurance data. You enable them to do so by changing an entry in their configuration (**pin.conf**) files. You make the same change in each configuration file.

To enable the utilities listed in Table 3-1 to generate revenue assurance data:

1. Open the utility configuration file in a text editor.
2. Set the **enable_ara** entry to **1**:

   ```
   - pin_mta enable_ara 1
   ```

   where **1** enables generation of revenue assurance data, and **0** disables generation of revenue assurance data.
3. Save and close the file.
4. Stop and start the Connection Manager (CM). See "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

# Configuring Custom Non-MTA Applications to Generate Revenue Assurance Data

You can configure your custom non-MTA application to generate revenue assurance data when it is run as a custom job in Business Operations Center. To do so, configure your custom application to do this:

1. Perform its custom processes.
2. Call the PCM_OP_PROCESS_AUDIT_CREATE opcode to create the **/process_audit/billing** object, passing these mandatory input fields in the opcode call:
   - PIN_FLD_ACCOUNT_OBJ
   - PIN_FLD_PROCESS_NAME
   - PIN_FLD_TOTAL_RECORDS
   - PIN_FLD_SUCCESSFUL_RECORDS
   - PIN_FLD_FAILED_RECORDS
3. Ensure that the status of the custom business operations job and your custom application was updated in the **/job/boc** object. To do so, call the PCM_OP_READ_FLDS opcode with the job ID passed as an argument and read the object's PIN_FLD_STATUS field.
4. If the **/job/boc** object's PIN_FLD_STATUS field has not been updated, update the field to the appropriate status:
   - **1**: The business operations job has been scheduled, but has not started yet.
   - **2**: The application has been started.
   - **3**: The application completed successfully.
   - **4**: The application failed.
   - **5**: The business operations job completed successfully.
   - **6**: The business operations job failed.
   - **7**: The business operations job could not run due to a scheduled blackout period.
   - **8**: The business operations job could not run because the system was down.


Copyright © 2017, 2026, Oracle and/or its affiliates.

# Enabling Billing Operations to Generate Revenue Assurance Data

In addition to the utilities in Table 3-1, the following operations can generate revenue assurance data:

- Bill Now
- Bill on purchase
- Auto-triggered billing

By default, however, these operations do not generate revenue assurance data.

To enable the operations to generate revenue assurance data:

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

   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```

   This command creates an XML file named **bus_params_billing.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_billing.xml.out**, set **EnableARA** to **enabled**:

   ```
   <EnableARA>enabled</EnableARA>
   ```

   > ⚠ **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_billing.xml.out** file to **bus_params_billing.xml**.
6. Use the following command to load your changes into the **/config/business_params** object:

   ```
   pin_bus_params bus_params_billing.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*.


Copyright © 2017, 2026, Oracle and/or its affiliates.

# Configuring Item Subtypes to Store Split Revenue Assurance Data

For bill items to work as configured in Revenue Assurance Manager, you must configure item subtypes.

Item subtypes define the different types of charges for item types such as cycle arrears, cycle forward, cycle forward arrears, and so on. For example, cycle forward fees and cycle arrears fees can be included in the **recurring** subtype and monthly usage in the **usage** subtype. For more information, see "About Splitting Revenue Assurance Data into Detailed Categories".

Before configuring item subtypes, you must configure bill items. See "Creating Custom Bill Items" in *BRM Configuring and Running Billing*.

To configure item subtypes:

1. Using a text editor, open the *BRM_home***/sys/data/pricing/example/config_item_types** file.
2. Define your new item subtypes in the **ItemType** tag.

   Use the item subtype **recurring** default item subtypes as examples. You can copy and change it as necessary.

   ```
   <ItemType precreate="false" type="cumulative" sub_type="recurring">/item/
   cycle_forward_arrear</ItemType>
   ```

   > **Note**
   >
   > Be sure to enter **Discount** for discount subtypes. Otherwise, Revenue Assurance Center displays the item's data in the Revenue section instead of the Adjustments/Discounts section.

3. In the Storable Class Editor, create the storable class **/item/***name*, where *name* is the value in the PIN_FLD_TYPE field of **/config/item_types**.
4. Run the "load_config_item_tags" and "load_config_item_types" utilities in *BRM Configuring and Running Billing* to configure the item subtypes in the BRM database.

All defined subtypes except discount subtypes are displayed in Revenue Assurance Center in the **Charge Type** column of the Billing Details screen under the **Revenue** section. Discount subtypes are displayed in the **Adjustments/Discounts** section of the Billing Details screen. For more information, see Revenue Assurance Center Help.


Copyright © 2017, 2026, Oracle and/or its affiliates.

4

# Collecting Revenue Assurance Data for Usage Write-Offs

Learn how to configure Oracle Communications Billing and Revenue Management (BRM) to collect revenue assurance data for usage write-offs.

Topics in this document:


- Configuring BRM to Collect Revenue Assurance Data for Usage Write-Offs


## Configuring BRM to Collect Revenue Assurance Data for Usage Write-Offs

For usage event records written off by Suspense Manager, Revenue Assurance Manager collects the original batch ID and the number of event records that were written off in the batch. See "About Collecting Revenue Assurance Data on Write-Offs".

To collect revenue assurance data for written-off usage event records, do the following:

- Configure event notification to call the PCM_OP_PROCESS_AUDIT_CREATE_WRITEOFF_SUMMARY opcode when Suspense Manager generates the **/event/notification/suspense/writeoff** event.

  See "Configuring Event Notification for Revenue Assurance".

- (Optional) Rename the default control point for collecting revenue assurance data on written-off usage event records.

  See "Renaming the Control Point for Collecting Data from Usage Write-Offs".

### Configuring Event Notification for Revenue Assurance

Revenue Assurance Manager uses event notification to collect data on written-off usage event records.

Before you can use Revenue Assurance Manager to collect such data, you must configure the event notification feature as follows:

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

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

2. Ensure that the merged file includes the entire event notification list in the *BRM_home***/sys/data/config/pin_notify_ra** file.

3. (Optional) If necessary to accommodate your business needs, add, modify, or delete entries in your final event notification list.

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

4. (Optional) If necessary to accommodate your business needs, create custom opcodes for event notification to trigger.

   See "Triggering Custom Operations" in *BRM Developer's Guide*.



Copyright © 2017, 2026, Oracle and/or its affiliates.

5. Load your final event notification list into the BRM database.

   See "Loading the Event Notification List" in *BRM Developer's Guide*.

> **Note**
>
> In addition, your business needs might require you to use event notification to call opcodes when other objects are created in the BRM database.

See "Using Event Notification" in *BRM Developer's Guide*.

# Renaming the Control Point for Collecting Data from Usage Write-Offs

The control point to collect revenue assurance data from written-off usage event records is configured in the CM configuration file.

To rename the control point for collecting data from usage write-offs:

1. Open the CM configuration file (*BRM_home***/sys/cm/pin.conf**) in a text editor.
2. Rename the control point in the **writeoff_control_point_id** entry.

   The default name for the control point is CP_SuspenseWriteOff.
3. Save and close the file.
4. Stop and restart the CM.


Copyright © 2017, 2026, Oracle and/or its affiliates.

5

# Using Batch IDs in Revenue Assurance Manager

Learn how to track event records with Oracle Communications Billing and Revenue Management (BRM) Revenue Assurance Manager using batch IDs during the recycling process.

Topics in this document:


- Tracking Event Records by Batch IDs


## Tracking Event Records by Batch IDs

Each batch file received from the mediation system is assigned a unique batch ID, which is stored in every record originating from that file. Each record also contains a distinct event ID.

During the recycling process, the record is assigned a new batch ID, but the original batch ID is preserved in an alternate field. This retention of the original batch ID allows for analyzing the revenue impact of event records for each batch received from mediation, even if some records undergo recycling.

Revenue Assurance Manager uses the following fields to monitor event records throughout the recycling process:

- DETAIL.BATCH_ID
- DETAIL.ORIGINAL_BATCH_ID
- DETAIL.ASS_SUSPENSE_EXT.SUSPENDED_FROM_BATCH_ID

For example, BRM receives a batch file with batch ID **Mediation087**. All event records associated with this file are assigned this batch ID. ECE processes event records from this batch and loads their data into the BRM database.

Later, some of the event records from this batch and a second batch, **Mediation099**, are recycled. During recycling, the two sets of event records from different batches are given the new batch ID **RecyclingBatch007**. When the individual event records are given the new batch ID, their original batch IDs are moved to the ORIGINAL_BATCH_ID field.

Table 5-1 contains selected data from a record in the batch after recycling.

**Table 5-1 Original Record Data**

| Event ID | Duration | Charge | Batch ID | Original Batch ID |
|---|---|---|---|---|
| 189 | 180 | 3 | Mediation087 | Mediation087 |

Table 5-2 contains the data for the record after recycling.



Copyright © 2017, 2026, Oracle and/or its affiliates.

**Table 5-2 Recycled Record Data**

| Event ID | Duration | Charge | Batch ID | Original Batch ID |
|---|---|---|---|---|
| 189 | 180 | 5 | RecyclingBatch007 | Mediation087 |

# Keeping Track of Rejected Event Records by Batch IDs

When a batch file is processed, BRM might be unable to rate some of its event records because of missing data or other reasons. The rejected event records can be processed by Suspense Manager and recycled.

When Suspense Manager recycles the event records, they receive new batch IDs based on the recycling batch. Their original batch IDs remain to reflect the mediation batch they started in. The suspended-from batch ID field (DETAIL.ASS_SUSPENSE_EXT.SUSPENDED_FROM_BATCH_ID) stores the ID of the batch in which the record was rejected.

For example, two batches (MED1 and MED2) are received from the mediation system and processed by Offline Mediation Controller. Some event records from each of the two batches are rejected and then recycled as part of batch RCL1. In addition, some event records from the original two batches are rerated as part of batch RRT1. Some of the event records in that rerating batch are rejected and then recycled as part of batch RCL2.

Table 5-3 summarizes how the three different batch ID fields change as event records are recycled.

**Table 5-3 Batch ID Changed Fields**

| Process | Value for BATCH_ID | Value for ORIGINAL_BATCH_ID | Value for SUSPENDED_FROM_BATCH_ID |
|---|---|---|---|
| Rating Batch MED1 | MED1 | MED1 | MED1[1] |
| Rating Batch MED2 | MED2 | MED2 | MED2[1] |
| Recycle Batch RCL1 (containing suspended event records from MED1 and MED2) | RCL1 | MED1/MED2 | MED1/MED2 |
| Rerating Batch RRT1 (containing event records from MED1 and MED2) | RRT1 | MED1/MED2 | RRT1[1] |
| Recycle Batch RCL2 (containing suspended event records from RRT1) | RCL2 | MED1/MED2 | RRT1 |

> **Note**
>
> The value of the suspended-from batch ID is ignored in rating and rerating. Because it is left blank, it is assigned the value of batch ID.


Copyright © 2017, 2026, Oracle and/or its affiliates.

6

# Configuring Revenue Assurance Alerts

Learn how to configure alerts based on revenue assurance data collected in your Oracle Communications Billing and Revenue Management (BRM) system.

Topics in this document:


- About Revenue Assurance Alerts
- Configuring Alert Thresholds
- Configuring Email Alerts
- Generating Revenue Assurance Alerts


## About Revenue Assurance Alerts

You can configure Revenue Assurance Manager to send alerts when the data collected meets criteria that you specify. For example, you can configure an alert to be sent when the number of event records rejected in one day exceeds a specified value.

Configuring revenue assurance alerts involves the following tasks:

- Configure alert thresholds in the **pin_config_ra_thresholds** file, and load them into **/config/ra_thresholds** objects.

  See "Configuring Alert Thresholds".

- Configure email alerts as follows:
  - Create the text of the email alerts in the **ra_alert_message.***locale* file, and load it into **/strings** objects. See "Specifying Email Alert Text".
  - Define email details, such as recipient information, in the **pin_config_ra_alerts** file, and load them into **/config/ra_alerts** objects. See "Specifying Email Alert Details".
- Check for alerts periodically by using the **pin_ra_check_thresholds** utility.

  See "Generating Revenue Assurance Alerts".

- Configure event notification to call the PCM_OP_PROCESS_AUDIT_POL_ALERT policy opcode when the **pin_ra_check_thresholds** utility generates the **/event/notification/ra_threshold** event.

  See "Configuring Event Notification for Revenue Assurance".

You can customize the default alert behavior by changing the PCM_OP_PROCESS_AUDIT_POL_ALERT opcode. For example, you could customize the opcode to log alert information in addition to sending emails. For details, see the chapter about billing in *BRM Opcode Guide*.

## Configuring Alert Thresholds

You configure alert thresholds in the **pin_config_ra_thresholds** file. When you configure an alert threshold, you define the following:

- The data you want to check and the time period in which it is checked.



Copyright © 2017, 2026, Oracle and/or its affiliates.

See "Specifying Threshold Data and Time Periods".

- Optional filter criteria to narrow results.

  See "Specifying Threshold Filter Conditions".

- Minimum and maximum threshold values for the data you specify.

  See "Specifying Threshold Field Value Ranges".

The values you specify work together to determine whether an alert is triggered. All conditions must be met for an alert to be triggered.

The **pin_config_ra_thresholds** file includes syntax information and examples. Consult the file. Use this section to supplement the information in the file.

After your alert thresholds are defined in **pin_config_ra_thresholds**, load that information into the database's **/config/ra_thresholds** objects. See "Loading Alert Details into the Database".

# Specifying Threshold Data and Time Periods

You specify an alert threshold's data and time periods in a line with the following syntax:

```
RA_Threshold:
<Threshold Name> <Obj Type> <Date Range Type> <WeekDay> <Number of Units> <Start Offset>
<Length of Period> <Secs/Mins/Hrs>
```

For example:

```
RA_Threshold:
Rejected_Records_Per_Day  /process_audit/batchstat/status  1  1  1  16:00:00  0  Secs
```

Table 6-1 provides explanations of the parameters in the definition.

**Table 6-1 Threshold Parameters**

| Parameter | Description |
|---|---|
| **Threshold Name** | Specifies the threshold name used when the alert is issued. The name should be as descriptive as possible. For example, **Duplicate_SMS_Records_Per_Week** is a good name for a weekly alert for the number of duplicate SMS records. |
| **Obj Type** | Specifies the **/process_audit** subclass containing the threshold value to check. For example, to generate alerts when the number of rejected event records reaches a certain value, enter **/process_audit/batchstat/status**. |
| **Date Range Type** | Specifies the time range covered by the alert:<br>• **Period alert:** Enter **0** to specify an alert configured for a specific time period. For example, you can configure a period alert to check data between 8:00 a.m. and 5:00 p.m. The time period is based on the process end time (PIN_FLD_PROCESS_END_T in the **/process_audit** object). Period alerts require a value for the **Length of Period** parameter. See "Period Alert Example".<br>• **Daily alert:** Enter **1** to specify a daily alert that repeats on a 24-hour basis. For example, you can configure a daily alert to check data every day from 4:00 p.m. on the current day to 4:00 p.m. the next day. See "Daily Alert Example".<br>• **Weekly alert:** Enter **2** to specify an alert that runs for an entire week. For example, you can configure a weekly alert to check data from 8:00 a.m. Monday to 8:00 a.m. the following Monday. See "Weekly Alert Example". |
| **WeekDay** | Specifies the day of the week on which a period or weekly alert is processed. Each day is represented by a numerical value between **1** (Sunday) and **7** (Saturday).<br>**Note:** This parameter does not apply to daily alerts. Set it to **0** for daily alerts. |


Copyright © 2017, 2026, Oracle and/or its affiliates.

**Table 6-1 (Cont.) Threshold Parameters**

| Parameter | Description |
|---|---|
| **Number of Units** | Specifies how many days or weeks a daily or weekly alert is processed. For example, if a daily alert must be processed for five days, enter **5**.<br>**Note:** This parameter does not apply to period alerts. Set it to **0** for period alerts. |
| **Start Offset** | Specifies the time of day that the alert starts. Enter the start time in hours, minutes, and seconds, expressed in 24-hour format. For example, to start an alert at 4:00 p.m., specify the start offset time as **16:00:00**. |
| **Length of Period** | Specifies the duration of period alerts, expressed in seconds, minutes, or hours (depending on the value specified in the **Secs/Mins/Hrs** parameter). The duration must be less than or equal to one week.<br>**Note:** This parameter applies only to period alerts. For other alerts, set it to **0**. |
| **Secs/Mins/Hrs** | Specifies the time unit that applies to the **Length of Period** parameter: seconds (**Secs**), minutes (**Mins**), or hours (**Hrs**). This value is ignored for daily and weekly alerts. |

# Specifying Threshold Filter Conditions

You set filter conditions by specifying required values for one or more fields in the **/process_audit** objects defined for the threshold. For example, if the threshold is checking **/process_audit/batchstat/status** objects, you could specify that the value of the PIN_FLD_EDR_STATUS field must be REJECTED.

You can enter more than one filter for a threshold. Data must meet all filter conditions.

You define filter conditions with the following syntax:

```
RA_Filter:
<Filter Field Name>  <Filter Int Value>  <Filter String Value>
<Filter Field Name>  <Filter Int Value>  <Filter String Value>
```

For example:

```
RA_Filter:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_STATUS  0  REJECTED
PIN_FLD_GROUP_DETAILS.PIN_FLD_SVC_TYPE  0  SMS
```

# Specifying Threshold Field Value Ranges

You specify threshold field value ranges in alerts so that when the thresholds are reached or crossed, the alert generates notifications. You set the ranges by specifying a minimum and maximum value for a field in the **/process_audit** objects defined for the threshold. For the threshold to be reached, the value of the field must be less than or equal to the minimum value *or* greater than or equal to the maximum value.

For example, if the threshold applies to **/process_audit/batchstat/status** objects, you could specify 0 and 75 as the minimum and maximum value for the PIN_FLD_EDR_COUNT field.

You can define multiple threshold value ranges for the same field. This enables you to send a new alert when the field value reaches a new threshold. For example, you could define an additional field value range for the PIN_FLD_EDR_COUNT field with a maximum of 150. This would trigger an alert if the value continues to rise after the initial alert at 75.

You define the threshold value range with the following syntax:


Copyright © 2017, 2026, Oracle and/or its affiliates.

```
RA_Field_Value_Range:
<Field Name>  <Max_Field_Value>  <Min_Field_Value>
<Field Name>  <Max_Field_Value>  <Min_Field_Value>
```

For example:

```
RA_Field_Value_Range:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_COUNT  50  0
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_COUNT  150 0
```

# Alert Threshold Examples

This section contains examples of the following threshold alert types:

- Period Alert Example
- Daily Alert Example
- Weekly Alert Example

## Period Alert Example

This period alert runs from 8 a.m. to 5 p.m. every Friday, and is triggered when the number of rejected event records exceeds 75 for that period:

```
RA_Threshold:
Rejected_Records_Per_Day  /process_audit/batchstat/status  0  6  0  08:00:00  9  Hrs

RA Filter:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_STATUS  0  REJECTED

RA_Field_Value_Range:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_COUNT  75  0
```

## Daily Alert Example

This alert runs daily starting at 4:00 p.m. It is triggered when the number of rejected event records reaches 1000 for the day. The alert is scheduled to repeat for 90 days.

```
RA_Threshold:
Rejected_Records_Per_Day  /process_audit/batchstat/status  1  0  90  16:00:00  0  Secs

RA Filter:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_STATUS  0  REJECTED

RA_Field_Value_Range:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_COUNT  1000  0
```

## Weekly Alert Example

This alert runs weekly starting at 8:00 a.m. on Monday. It is triggered when the number of rejected SMS records reaches 100 during that period and again when the number reaches 250. It is valid for an entire year (52 weeks).

```
RA_Threshold:
Reject_SMS_EDRs_Week  /process_audit/batchstat/
svctype_revstream  2  2  52  08:00:00  0  Secs

RA Filter:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_STATUS  0  REJECTED
PIN_FLD_GROUP_DETAILS.PIN_FLD_SVC_TYPE  0  SMS
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

```
RA_Field_Value_Range:
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_COUNT  100  0
PIN_FLD_GROUP_DETAILS.PIN_FLD_EDR_COUNT  250  0
```

## Loading Alert Details into the Database

After you configure alert details, use the "load_pin_config_ra_thresholds" utility to load them into the **/config/ra_thresholds** object. The "pin_ra_check_thresholds" utility then uses the threshold information in the **/config/ra_thresholds** object to search revenue assurance data when it generates alerts.

### Loading Alert Details by Using load_pin_config_ra_thresholds

To load your threshold values into the BRM database:

> **Note**
>
> The **load_pin_config_ra_thresholds** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*. If the utility fails, it does not overwrite the existing data.

1. Open the *BRM_home***/sys/data/config/pin_config_ra_thresholds** file in a text editor.
2. Use the instructions and examples in **pin_config_ra_thresholds** to configure alert details.
3. Save and close the file.
4. Load the alert details into the database by using this command:

   ```
   load_pin_config_ra_thresholds pin_config_ra_thresholds
   ```

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

   ```
   load_pin_config_ra_thresholds BRM_home/sys/data/config/pin_config_ra_thresholds
   ```

   > **Tip**
   >
   > If you copy the **pin_config_ra_thresholds** file to the directory from which you run the **load_pin_config_ra_thresholds** utility, you do not have to specify the path or file name. The file must be named **pin_config_ra_thresholds**.

   See "load_pin_config_ra_thresholds".

   To verify that the alert details were loaded, display the **/config/ra_thresholds** object by using Object Browser or by using the **robj** command with the **testnap** utility. See "Reading an Object and Writing its Contents to a File" in *BRM Developer's Guide*.

# Configuring Email Alerts

Configuring email alerts involves the following tasks:

- Specify the text you want in alert emails, and optionally localize that text.


Copyright © 2017, 2026, Oracle and/or its affiliates.

See "Specifying Email Alert Text".

- Specify email details, and load them into the database.

  See "Specifying Email Alert Details".

# Specifying Email Alert Text

You configure the subject and message body of email alert messages in the locale of your choice. To configure email alert messages in U.S. English, edit the *BRM_home***/sys/msgs/revenueassurance/ra_alert_message.en_US** file. This file includes a default subject line and message body that you can edit. The file also includes information about syntax and required elements, which follow standard rules for localized string files. See "Localizing and Customizing Strings" in *BRM Developer's Guide*.

You can use the following substitution parameters in the file. They are replaced with appropriate information when the alert email is sent.

- %ALERT_NAME: Replaced with the alert name.
- %START_TIME: Replaced with the start time of the period for which alert was generated.
- %END_TIME: Replaced with the end time of the period for which the alert was generated.
- %OBJECT_TYPE: Replaced with the object type value configured for the alert.
- %FILTERS: Replaced with the field-value pairs of the filter conditions defined for the alert.
- %THRESHOLD_VALUES: Replaced with the field and corresponding minimum and maximum limits defined for the alert.
- %LINE_BREAK: Replaced with a new line. (Applicable to the message body only.)

The default subject line is the following:

```
Alert %ALERT_NAME has crossed defined threshold values.
```

The default message body is the following:

```
The alert %ALERT_NAME has crossed defined threshold values for the period %START_TIME to
%END_TIME. %LINE_BREAK The alert is configured as given below: %LINE_BREAK ObjectType =
%OBJECT_TYPE %LINE_BREAK Filter = %FILTERS %LINE_BREAK Threshold Values =
%THRESHOLD_VALUES"
```

You can define localized versions of the email alert text by creating **ra_alert_message.***locale* files with the appropriate file name extension for the locales. See "Creating New Strings and Customizing Existing Strings" in *BRM Developer's Guide*.

After defining the email text, use the **load_localized_strings** utility to load the contents of the file into the **/strings** objects.

The load command syntax is as follows:

```
load_localized_strings ra_alert_message.locale
```

See "load_localized_strings" in *BRM Developer's Guide*.

> **Note**
>
> When loading a localized version of the file, use the correct file extension for your locale. For a list of file extensions, see "Locale Names" in *BRM Developer's Guide*.


Copyright © 2017, 2026, Oracle and/or its affiliates.

See "Loading Localized or Customized Strings" in *BRM Developer's Guide*.

# Specifying Email Alert Details

You specify recipients and other details about email alerts in the **pin_config_ra_alerts** file. Specify the following details for each alert:

- Name of the alert
- Locale to be used for subject and email body

  The locale determines which localized version of the alert text is sent.
- Name of the sender
- Email addresses of recipients

Specify the sender name and locale for each alert in a single line with the following syntax:

```
RA_Alert: Alert_Name Sender_Name Locale
```

> **Note**
>
> Each entry must begin with **RA_Alert**; the text is case-sensitive.

The **Alert_Name** parameter must be the name of an alert as defined in the **pin_config_ra_thresholds** file. The sender name is the name that appears in the **From** line of email messages.

You define recipients for each alert by specifying email addresses and types in a block below the **RA_Alert** line. Use the following syntax:

```
Email_addr_1 Email_Type>
Email_addr_2 Email_Type>
Email_addr_3 Email_Type>
```

The email type is optional. It determines how the email is sent to the recipient. The following are valid values:

- **1:** Email is addressed directly to the recipient.
- **2:** The recipient receives a copy (CC).
- **3:** The recipient receives a blind copy and does not see other recipients (BCC).
- **4:** Specifies that the address is to be used as the Reply To value in the alert message.

  This address is not a recipient of the alert email.

The default value is **1**, which is used if no type is specified.

The following example configures email notification for the Rejected_Records_Per_Day alert. The sender of alert emails is RA_NOTIFICATION. The primary recipient is RA_admin@mailserver.com, and RA_operator@mailserver.com receives a copy.

```
RA_Alert:  Rejected_Records_Per_Day  RA_NOTIFICATION  en_US

RA_admin@mailserver.com  1
RA_operator@mailserver.com  2
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

For more information, see the *BRM_home*/**sys/data/config/pin_config_ra_alerts** file, which includes syntax information and examples.

After configuring alert details, load the file into the database by using the "load_pin_config_ra_alerts" utility. Email alert data is stored in the **/config/ra_alerts** object.

> **Note**
>
> The **load_pin_config_ra_alerts** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*. If the utility fails, it does not overwrite the existing data.

To configure and load email alert details:

1. Open the *BRM_home*/**sys/data/config/pin_config_ra_alerts** file in a text editor.
2. Add your email details to the file. The file includes instructions and sample data.
3. Save and close the file.
4. Load the email details into the database by using this command:

   ```
   load_pin_config_ra_alerts pin_config_ra_alerts
   ```

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

   ```
   load_pin_config_ra_alerts BRM_home/sys/data/config/pin_config_ra_alerts
   ```

   > **Tip**
   >
   > If you copy the **pin_config_ra_alerts** file to the directory from which you run the **load_pin_config_ra_alerts** utility, you do not have to specify the path or file name. The file must be named **pin_config_ra_alerts**.

   See "load_pin_config_ra_alerts".

To verify that the email details were loaded, display the **/config/ra_alerts** object by using Object Browser or by using the **robj** command with the **testnap** utility. See "Reading an Object and Writing its Contents to a File" in *BRM Developer's Guide*.

# Generating Revenue Assurance Alerts

You use the "pin_ra_check_thresholds" utility to check alert thresholds periodically. You can run **pin_ra_check_thresholds** as part of a **cron**, **crontab**, or **at** script, or you can run it manually.

This utility checks the **/config/ra_thresholds** object for the revenue assurance thresholds you set. If it finds that a threshold has been crossed, it generates a notification event that causes the PCM_OP_PROCESS_AUDIT_POL_ALERT policy opcode to prepare and send an email alert. You can customize the opcode to change the default alert behavior. See the chapter about billing in *BRM Opcode Guide*.


Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> The **pin_ra_check_thresholds** utility needs a configuration (**pin.conf**) file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

Use the following command to run the **pin_ra_check_thresholds** utility:

```
pin_ra_check_thresholds
```

See "pin_ra_check_thresholds".


Copyright © 2017, 2026, Oracle and/or its affiliates.

7

# Revenue Assurance Utilities

Learn about the syntax and parameters for the Oracle Communications Billing and Revenue Management (BRM) revenue assurance utilities.

Topics in this document:


- load_ara_config_object
- load_pin_config_ra_alerts
- load_pin_config_ra_flows
- load_pin_config_ra_thresholds
- pin_ra_check_thresholds


## load_ara_config_object

Use the **load_ara_config_object** utility to specify how to check for duplicate audit objects. This utility loads data into the **/config/process_audit_conf** object in the BRM database.

> ⚠ **Caution**
>
> The **load_ara_config_object** utility overwrites existing **/config/process_audit_conf** objects. If you are updating existing configuration data, you cannot load new configuration data only. You must load a complete set of configuration data each time you run the **load_ara_config_object** utility.

> ⓘ **Note**
>
> To connect to the BRM database, the **load_ara_config_object** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

### Location

*BRM_home***/bin**

### Syntax

**`load_ara_config_object`** *`pin_config_audit_conf_file`*



Copyright © 2017, 2026, Oracle and/or its affiliates.

### Parameters

***pin_config_audit_conf_file***
The name and location of the file that defines the **/process_audit** object fields used for performing duplicate checks. By default, the utility uses the *BRM_home***/sys/data/config/pin_config_audit_conf** file.

> **Tip**
>
> The file must be named **pin_config_audit_conf**.

If you run the command in a different directory from where the **pin_config_audit_conf** file is located, you must include the entire path for the file.

### Results

The **load_ara_config_object** 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.

# load_pin_config_ra_alerts

Use the **load_pin_config_ra_alerts** utility to configure email details in the **/config/ra_alerts** object in the BRM database. You define the email details in the **pin_config_ra_alerts** file in *BRM_home***/sys/data/config**. See "Configuring Email Alerts".

> **Caution**
>
> The **load_pin_config_ra_alerts** utility overwrites existing **/config/ra_alerts** objects. If you are updating revenue assurance alerts, you cannot load new alerts only. You must load complete sets of alerts each time you run the **load_pin_config_ra_alerts** utility.

> **Note**
>
> To connect to the BRM database, the **load_pin_config_ra_alerts** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

### Location

*BRM_home***/bin**

### Syntax

```
load_pin_config_ra_thresholds ra_alerts_file
```


Copyright © 2017, 2026, Oracle and/or its affiliates.

### Parameters

***ra_alerts_file***
The name and location of the file that contains the configured alert. The default **pin_config_ra_alerts** file is in *BRM_home***/sys/data/config**.
The **pin_config_ra_alerts** file contains the rules for configuring email details and includes an example. You can customize this example and configure your email details.
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_ra_thresholds BRM_home/sys/data/config/pin_config_ra_alerts
```

> **Tip**
>
> If you copy the **pin_config_ra_alerts** file to the directory from which you run the **load_pin_config_ra_alerts** utility, you do not have to specify the path or file name. The file must be named **pin_config_ra_alerts**.

### Results

The **load_pin_config_ra_alerts** utility notifies you when it successfully creates the **/config/ra_alerts** 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 the alerts were loaded, display the **/config/ra_alerts** object by using the Object Browser or by using the **robj** command with the **testnap** utility. See "Reading an Object and Writing its Contents to a File" in *BRM Developer's Guide*.

# load_pin_config_ra_flows

Use the **load_pin_config_ra_flows** utility to load revenue assurance flows into the **/config/ra_flows** object in the BRM database. You define the flows in the **pin_config_ra_flows** file in *BRM_home***/sys/data/config**.

> **Caution**
>
> The **load_pin_config_ra_flows** utility overwrites existing flows. If you are updating flows, you cannot load new flows only. You must load complete sets of flows each time you run the **load_pin_config_ra_flows** utility.

> **Note**
>
> To connect to the BRM database, the **load_pin_config_ra_flows** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

### Location

*BRM_home***/bin**


Copyright © 2017, 2026, Oracle and/or its affiliates.

### Syntax

```
load_pin_config_ra_flows ra_flows_file
```

### Parameters

***ra_flows_file***
The name and location of the file that contains the configured flows. The default **pin_config_ra_flows** file is in *BRM_home***/sys/data/config**.
The **pin_config_ra_flows** file contains the rules for configuring flows and includes an example. You can customize this example and configure flows.
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_ra_flow BRM_home/sys/data/config/pin_config_ra_flows
```

> **Tip**
>
> If you copy the **pin_config_ra_flows** file to the directory from which you run the **load_pin_config_ra_flows** utility, you do not have to specify the path or file name. The file must be named **pin_config_ra_flows**.

### Results

The **load_pin_config_ra_flows** utility notifies you when it successfully creates the **/config/ra_flows** 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 the flows were loaded, display the **/config/ra_flows** object by using the Object Browser or by using the **robj** command with the **testnap** utility. See "Reading an Object and Writing its Contents to a File" in *BRM Developer's Guide*.

# load_pin_config_ra_thresholds

Use the **load_pin_config_ra_thresholds** utility to load revenue assurance alert threshold details into the **/config/ra_thresholds** object in the BRM database. You define these thresholds in the **pin_config_ra_thresholds** file in *BRM_home***/sys/data/config**. See "Configuring Alert Thresholds".

> **Caution**
>
> The **load_pin_config_ra_thresholds** utility overwrites existing thresholds. If you are updating revenue assurance thresholds, you cannot load new thresholds only. You must load complete sets of thresholds each time you run the **load_pin_config_ra_thresholds** utility.


Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> To connect to the BRM database, the **load_pin_config_ra_thresholds** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*.

### Location

*BRM_home*/**bin**

### Syntax

```
load_pin_config_ra_thresholds ra_thresholds_file
```

### Parameters

***ra_thresholds_file***
The name and location of the file that contains the configured threshold details. The default **pin_config_ra_thresholds** file is in *BRM_home*/**sys/data/config**.
The **pin_config_ra_thresholds** file contains the rules for configuring thresholds and includes an example. You can customize this example and configure thresholds.
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_ra_thresholds BRM_home/sys/data/config/pin_config_ra_thresholds
```

> **Note**
>
> If you copy the **pin_config_ra_thresholds** file to the directory from which you run the **load_pin_config_ra_thresholds** utility, you do not have to specify the path or file name. The file must be named **pin_config_ra_thresholds**.

### Results

The **load_pin_config_ra_thresholds** utility notifies you when it successfully creates the **/config/ra_thresholds** 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 the alert thresholds were loaded, display the **/config/ra_thresholds** object by using the Object Browser or by using the **robj** command with the **testnap** utility. See "Reading an Object and Writing its Contents to a File" in *BRM Developer's Guide*.

# pin_ra_check_thresholds

Use the **pin_ra_check_thresholds** utility to notify you when the **/config/ra_thresholds** object contains revenue assurance alerts. Alerts are loaded into this object when the revenue assurance thresholds you configured are crossed. See "Configuring Alert Thresholds".


Copyright © 2017, 2026, Oracle and/or its affiliates.

> **Note**
>
> To connect to the BRM database, the **pin_ra_check_thresholds** utility needs a configuration file in the directory from which you run the utility. See "Connecting BRM Utilities" in *BRM System Administrator's Guide*. You can change the configuration **logfile** and **loglevel** entries as needed.

### Location

*BRM_home***/bin**

### Syntax

```
pin_ra_check_thresholds
```

### Parameters

There are no parameters for the **pin_ra_check_thresholds** utility.

### Results

The **pin_ra_check_thresholds** utility returns the values listed in "Return Values" and logs other information in the revenue assurance log file. This log file is either in the directory from which the utility was started or in a directory specified in the utility configuration file.

### Return Values

**pin_ra_check_thresholds** returns the following:

- **0:** Success
- **1:** Error opening the PCM connection
- **2:** The **/config/ra_thresholds** object is not present
- **3:** All other errors


Copyright © 2017, 2026, Oracle and/or its affiliates.