Oracle® Communications Billing and Revenue Management
Patch Set Release Notes
Contents

Preface

Audience ix
Accessing Oracle Communications Documentation ix
Documentation Accessibility ix
Document Revision History ix

1 Summary of Customer-Reported Fixes

Customer-Reported Fixes in BRM 1-1
Customer-Reported Fixes in ECE 1-21
Customer-Reported Fixes in PDC 1-23
Customer-Reported Fixes in Billing Care 1-26
Customer-Reported Fixes in Business Operations Center 1-31

2 New Features in BRM

New Features in BRM 12.0 Patch Set 2 2-1
  Deploying BRM Services on a Cloud Native Environment 2-1
  Wholesale Billing Enhancements 2-2
    Specifying How to Calculate Deferred Taxes for Wholesale Billing 2-2
  Additional Card Security Presence Values Supported for Card Validation or Authorization 2-3
  Card Security Code Is Now Masked in Logs 2-4
  Display Bills Generated Before Moving the Account to a Hierarchy 2-4
  Managing Discount Validity Starting or Ending in Future Cycles 2-5
  Wildcard in Item Type Selectors 2-5
  BRM Supports POID Generation in ECE 2-6
    Enabling Prepaid-Event Partitions 2-6
    About Creating Custom Classes 2-7
  Support for Stored-Credential Transactions for Payments 2-8
    Storing Card Credentials for Future Transactions 2-10
    Purging Card Credentials 2-11
    Migrating Legacy Payment Information 2-11
Support for Migration of Legacy Data into BRM and ECE in Real Time
  Importing Data 2-13
  Deploying Data in BRM and ECE 2-14
  Merging Legacy Balances 2-15
Support for Migrating Hierarchical Accounts Using Same Input File 2-16
Event Rounding Rules Can Be Used for Adjustments 2-16
New Staging Directory for Exported G/L reports 2-16
BRM Client Applications Supported from 12.0 Patch Set 2 2-17
New Features in BRM 12.0 Patch Set 1 2-17
Improved Performance for Large Accounts 2-18
  Enabling Wholesale Billing 2-18
  Creating Wholesale Accounts and Bill Unit Hierarchies 2-19
  Configuring Wholesale Business Profile 2-20
  Setting Up a Wholesale Parent 2-21
  Rolling Charges Up to the Wholesale Parent 2-21
  Rolling A/R Actions Up to the Wholesale Parent 2-22
  Rolling Journals Up to the Wholesale Parent 2-22
  Running Wholesale Billing 2-23
  Configuring Billing Delay for Wholesale Hierarchies 2-24
  Setting Up Billing-Time Discounts for Wholesale Hierarchies 2-24
  Suppressing Bills for Wholesale Hierarchies 2-24
  Trial Billing for Wholesale Hierarchies 2-24
  Support for A/R Activities 2-25
  Specifying Search Criteria for Retrieving Items, Events, and Bills 2-25
  Moving Bill Units into or out of Wholesale Hierarchies 2-26
  Configuration Changes 2-26
  Opcode Changes 2-26
  Schema Changes 2-27
  Storable Class Changes 2-27
  Utility Changes 2-28
  pin_update_journal 2-28
  pin_roll_up_ar_items 2-29
Delay Interval Can be Configured for Resolving Failed Payments 2-29
Enhanced Data Protection 2-30
  Specifying Retention Period for Closed Accounts 2-31
  Deleting Closed Accounts 2-31
  pin_del_closed_accts 2-33
Enhanced Security for Root Wallet 2-34
Support for Rolling Back the BRM Patch Set 2-34
BRM 12.0 Is Now Certified with Mozilla Firefox 58.0 2-34
BRM 12.0 Is Now Certified with Perl 5.28.0 2-34
New Features in ECE

- New Features in ECE 12.0 Patch Set 2
- New Tools to Monitor ECE
  - Running ECE Monitoring Tools
  - Customizing JMX-Based Tool Reports
  - Monitoring ECE Cluster
  - Monitoring ECE Using Grafana and Prometheus
  - Starting and Stopping ECE Metrics Collector
  - Collecting ECE Metrics Data
  - Setting Log Levels by Using Scripts
  - Using cohql_file_processor for Query
  - Duplicate Check Enhancement in ECE
    - Charging Operation Type Can be Configured for Expired Active Session Cleanup
      - Configuring Expired Objects Cleanup
    - ECE Now Generates POID for Events
      - Configuring Cluster ID
      - Connecting Rated Event Formatter Instances
      - Enabling Prepaid-Event Partitions
    - New Tool for Querying ECE Cache Data
    - ECE Now Supports Wildcard in Item Type Selectors
    - Support for Persisting BRS Configuration Data
    - Rated Event Partition in the ECE Persistent Database

New Features in ECE 12.0/12.0 Patch Set 1

New Features in PDC

- New Features in PDC 12.0 Patch Set 2
- Support for Configuring Item Type Selectors in the PDC User Interface
- PDC Web Services for Creating and Exporting Components
  - Creating or Modifying Components using PDC Web Services
  - Exporting Components using PDC Web Services
- PDC Synchronizes Event Data Using Event Types
  - Synchronizing and Publishing Event Type to ECE
- PDC Now Supports Wildcard in Item Type Selectors
- Support for Enabling SSO Using SAML

New Features in PDC 12.0 Patch Set 1
5  New Features in Billing Care

New Features in Billing Care 12.0 Patch Set 2  5-1
  SDK Enhancements for Customizing Billing Care  5-1
  View Bills Generated Before Moving the Account to a Hierarchy  5-2
  Support for Account Number and Bill Number in Billing Care Embeddable URLs  5-2
  Embeddable URLs for Purchase Package and Purchase Bundle Screens  5-2

New Features in Billing Care 12.0 Patch Set 1  5-2
  Additional Options for Searching Bills  5-3
  Billing Care Is Now Certified with Mozilla Firefox 58.0  5-3
  Billing Care is Now Certified with Oracle Identity and Access Management 12c Release 2  5-3
  SDK for Customizing the Purchase Flow and Assets Display  5-3
  Support for Backdated Account Creation and Product Purchase  5-3
  Support for Creating Wholesale Accounts and Billing Hierarchies  5-3
  Support for Customizing Product Information Post Account Creation  5-4
  Support for Deferred Actions  5-4
  Support for Multischema  5-4
  Support for Undefined Payment Method  5-4
  Support for Enabling SSO using SAML  5-4

6  New Features in Business Operations Center

New Features in Business Operations Center 12.0 Patch Set 2  6-1
  Workflow Job to Automate Billing Process  6-1

New Features in Business Operations Center 12.0 Patch Set 1  6-2
  Business Operations Center is Now Certified with Jersey Bundle 2.27 (Patch Set 1)  6-3
  Business Operations Center is Now Certified with Jackson 2.9.6 (Patch Set 1)  6-3
  Business Operations Center is Now Certified with Log4j2 2.11.1 (Patch Set 1)  6-3
  Business Operations Center Is Now Certified with Mozilla Firefox 58.0 (Patch Set 1)  6-3
  Business Operations Center is Now Certified with Oracle Identity and Access Management 12c Release 2 (Patch Set 1)  6-3
  Business Operations Center is Now Certified with Quartz Scheduler 2.3.0 (Patch Set 1)  6-3

7  Opcode Changes

New Opcodes  7-1
8 Storable Class Changes

New Storable Classes 8-1
Changed Storable Classes 8-1

9 Schema and Index Changes

BRM Schema and Index Changes 9-1
  Schema and Index Changes (Patch Set 2) 9-1
  Schema and Index Changes (Patch Set 1) 9-3

10 Utility Changes

BRM Utilities 10-1
  New Utilities 10-1
  Changed Utilities 10-1
  Removed Utilities 10-2

11 Known Problems

Known Problems in BRM 11-1
  A/R Actions between Billing and Invoicing Not Captured in Invoice 11-1
  Bill Now Generates Two Audit Objects for Nonpaying Child Items 11-1
  Default Invoices Do Not Display Custom Events and Items 11-1
  Error When Configuring or Modifying Security Policies for Web Services Manager for JAX-WS 11-1
  Event Browser Displays Data Internal to BRM in the Event Description 11-2
  Invoices Do Not Display Charge Sharing Details Separately 11-2
  Invoices Might Display Wrong Discount Amount 11-2
  pin_inv_accts Fails for Migrated Accounts if Business Intelligence Publisher Invoicing Integration Is Enabled 11-2
  Refund Is Incorrectly Calculated When Remaining Charge Discount Canceled 11-3
  Sample Prepaid Service Life Cycle Does Not Support Provisioning of Telco Services 11-3
  The SLM Feature Does Not Support Account Billing Relationships 11-3
Settlement Taxes Might Be Incorrectly Calculated 11-4
stop_eai.js Does Not Work 11-4
Summary Invoices Do Not Show Real-Time Discount and Tax Details for Items 11-4
Tax Not Recalculated after Rerating 11-5
Known Problems in Business Operations Center 11-5
Known Problems in Billing Care 11-5
  Duplicate Login Request When Using Single Sign-On 11-5
  Firefox and Internet Explorer Fail to Connect Securely to Billing Care 11-6
Known Problems in PDC 11-6
  PDC Search Functionality Not Working As Expected 11-6
  Migrating Pricing Data Fails in a PDC System with BRE and RRE 11-6
  Adding Balance Groups and Setting Credit Limits in a Package Displays Exceptions 11-7
  Migration Fails Even When the Migration Process is Enabled to Skip Errors 11-7
  You Are Not Warned to Save Your Changes 11-7
Known Problems in PCC 11-7
Preface

This document includes information and the detailed information on the patch set features about Oracle Communications Billing and Revenue Management (BRM) 12.0 patch sets.

Audience

This document is intended for all BRM users.

Accessing Oracle Communications Documentation

BRM documentation and additional Oracle documentation, such as Oracle Database documentation, is available from Oracle Help Center:

http://docs.oracle.com

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

https://edelivery.oracle.com

Documentation Accessibility

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

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Document Revision History

The following table lists the revision history for this guide:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F10044-01</td>
<td>September 2018</td>
<td>Initial release. Added information about Patch Set 1 features, which are identified by &quot;(Patch Set 1).&quot;</td>
</tr>
<tr>
<td>Version</td>
<td>Date</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>F10044-02</td>
<td>September 2019</td>
<td>Added information about Patch Set 2 features, which are identified by &quot;(Patch Set 2).&quot;</td>
</tr>
</tbody>
</table>
Summary of Customer-Reported Fixes

This chapter provides an overview of the customer-reported bug fixes that were introduced in Oracle Communications Billing and Revenue Management (BRM) 12.0 Patch Set 1 and Patch Set 2.

Customer-Reported Fixes in BRM

Table 1-1 lists the service request (SR) issues reported by external sources for BRM and provides a brief description of the resolution. The SRs are grouped by the respective patch sets.

Table 1-1  Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-15091245471</td>
<td>26618794</td>
<td>(Patch Set 2) In order to avoid duplicate tax calculation and incorrect tax adjustments, the <code>is_tax_calculated</code> flag was introduced earlier for pipeline rated events. But this caused an issue in the tax calculation during the event adjustment, when the event included had more than one tax balance impact. This has been fixed.</td>
</tr>
<tr>
<td>3-15286560801</td>
<td>27123464</td>
<td>(Patch Set 2) It was not possible to set the promise-to-pay date as the system date. This has been fixed. Now, additional validations are added to ensure that the promise-to-pay date is greater than the system date.</td>
</tr>
<tr>
<td>3-15853592531</td>
<td>27632734</td>
<td>(Patch Set 2) Paymentech Data Manager (DM) now supports Customer-Initiated Transaction (CIT) and Merchant-Initiated Transaction (MIT) for processing VISA credit cards. See &quot;Support for Stored-Credential Transactions for Payments&quot;.</td>
</tr>
<tr>
<td>3-17434540731</td>
<td>27993355</td>
<td>(Patch Set 2) The POID_ID required an increase in the number of bits from its current value of 44 bits because the allocated range of Portal object IDs (POID) were not sufficient. This has been now increased to 51 bits.</td>
</tr>
<tr>
<td>3-17757279581</td>
<td>28342613</td>
<td>(Patch Set 2) A payment of zero amount was failing in BRM with an error because there was PIN_FLD_PAYMENT value was not set in the response. This has been fixed.</td>
</tr>
<tr>
<td>3-17757279623</td>
<td>28387497</td>
<td>(Patch Set 2) BRM was not providing flexibility to set the transaction type for payments processed through Paymentech DM in the PCM_OP_PYMT_POL_PRE_COLLECT policy opcode. This has been fixed.</td>
</tr>
<tr>
<td>3-17747525701</td>
<td>28444532</td>
<td>(Patch Set 2) Oracle Data Manager was incorrectly logging an error message when insufficient back ends were configured. This has been fixed and now a warning message is displayed.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-17796627641</td>
<td>28466602</td>
<td>(Patch Set 2) Indexes were rebuilt every time the audit tables were purged. This has been fixed. The <code>purge_audit_tables.pl</code> script has been modified and the indexes are not rebuilt if the <code>$rebuild_index</code> entry in the <code>BRM_home/sys/archive/oracle/purge_audit_tables.conf</code> file is set to <code>no</code>, where <code>BRM_home</code> is the directory in which you installed BRM components. If this entry is not set or set to any other value, the indexes are rebuilt every time the audit tables are purged.</td>
</tr>
<tr>
<td>3-17732092221</td>
<td>28490340</td>
<td>(Patch Set 2) During the billing process, the end date in a cycle arrear event was set incorrectly when the product end date was the same as the <code>actg_next_t</code> value. This caused the resources to have incorrect validity dates. This has been fixed.</td>
</tr>
<tr>
<td>3-18370747541</td>
<td>28521329</td>
<td>(Patch Set 2) Paymentech DM now supports CIT and MIT for processing Mastercard, Japan Credit Bureau (JCB), and Diners Club credit cards. This has been fixed.</td>
</tr>
<tr>
<td>3-18039304221</td>
<td>28543518</td>
<td>(Patch Set 2) During subscription transfer, Adaptec AdvancedRAID Controller (AAC) details (for example, <code>PIN_FLD_AAC_PROMO_CODE</code>) of the service from the source account were not copied to the new service created at the destination account. This has been fixed.</td>
</tr>
<tr>
<td>3-18126737181</td>
<td>28549500</td>
<td>(Patch Set 2) The <code>pin_recover</code> utility was failing while searching for payments to be recovered. This has been fixed.</td>
</tr>
<tr>
<td>3-17564929691</td>
<td>28552543</td>
<td>(Patch Set 2) Web Services Manager was always sending the date-time stamp information in Universal Time Coordinated (UTC) as part of the response without considering the actual timezone. This has been fixed to send the date-time stamp as per the time zone configuration in the <code>BRM_home/deploy/web_services/Infranet.properties</code> file.</td>
</tr>
<tr>
<td>3-17790987201</td>
<td>28556710</td>
<td>(Patch Set 2) The rerate process did not update the original event. This has been fixed and some additional criteria has been added to identify and update the corresponding original event during rerating.</td>
</tr>
<tr>
<td>3-17612813861</td>
<td>28588801</td>
<td>(Patch Set 2) During the sequential discounting, non-applicable events were also considered while calculating the periods. This has been fixed.</td>
</tr>
<tr>
<td>3-17757279623</td>
<td>28596194</td>
<td>(Patch Set 2) The <code>fm_pymt</code> module was not sending transaction type to Paymentech DM. This has been fixed. Now, transaction type is used by Paymentech DM to identify the type of payment, for example, recurring, installment, and so on.</td>
</tr>
<tr>
<td>3-17697487631</td>
<td>28603124</td>
<td>(Patch Set 2) The scripts in <code>BRM_home/bin</code> was using two different versions of HTTPClient libraries, 4.4 and 4.5.3. This has been fixed to use only 4.5.3 version.</td>
</tr>
</tbody>
</table>

Table 1-1  (Cont.) Customer-Reported Fixes for BRM
### Table 1-1 (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-17223827112</td>
<td>28607064</td>
<td>(Patch Set 2) In a BRM deployment with Oracle Communications Elastic Charging Engine (ECE) as charging engine, when the BRM Oracle Data Manager commit failed, the BRM data was not in synchronization with the ECE cache data. This has been fixed by triggering a commit to ECE only after the BRM Oracle Data Manager commit is successful.</td>
</tr>
<tr>
<td>3-18187073311</td>
<td>28631061</td>
<td>(Patch Set 2) In the subscription opcode PCM_OP_SUBSCRIPTION_SET_PRODINFO, a memory leak was observed. This has been fixed.</td>
</tr>
<tr>
<td>3-17444287241</td>
<td>28638444</td>
<td>(Patch Set 2) In the subscription module, during the product cancellation, an incorrect scale for refund was used. This has been fixed.</td>
</tr>
<tr>
<td>3-17803665231</td>
<td>28671913</td>
<td>(Patch Set 2) The EVENT_POID_LIST required to store longer strings than it was allowed (4000 characters) in the database. This has been fixed.</td>
</tr>
<tr>
<td>3-18188014831</td>
<td>28678759</td>
<td>(Patch Set 2) A memory leak was observed in flist to string conversion routine. This has been fixed.</td>
</tr>
<tr>
<td>3-17547976531</td>
<td>28689682</td>
<td>(Patch Set 2) During rerating, the impact type for PIN_IMPACT_TYPE_TAX_RERATED was not being set to PIN_IMPACT_TYPE_TAX, and this was leading to incorrect tax amounts in the invoices. This has been fixed.</td>
</tr>
<tr>
<td>3-17920861571</td>
<td>28700838</td>
<td>(Patch Set 2) Backdated adjustments were not displayed in the corrective invoices. This has been fixed.</td>
</tr>
</tbody>
</table>
| 3-1772323998   | 28774361    | (Patch Set 2) The **PaymentDeallocation** feature allows bill adjustment on a zero due bill or a partially paid bill. The same behavior has been now implemented for item and event-level adjustments. If **BillPaymentDeallocation** is enabled, do the following to allow the item and event-level adjustments:  
   1. Unallocate the payment applied on the corresponding bill item.  
   2. Do the item or event-level adjustment.  
   3. Reallocate the payment back to the bill item.  
   If the payment amount is more than the due amount after the adjustment is done, then some unallocated amount is stored in *item/payment*. This unallocated amount is displayed in the *Adjustment/Payments not applied* tab of Customer Center. |
| 3-17519084381  | 28774366    | (Patch Set 2) If an error is thrown in ECE, the error was always set to PIN_ERR_BAD_VALUE. Hence, it was not possible to identify the actual error details. This has been fixed.  
   In case the output flist of the PUBLISH_ECE_EVENT operation contains ERROR_DESCR set to **Credit Ceiling Breached**, the ebuf is set to PIN_ERR_CREDIT_LIMIT_BREACHED instead of PIN_ERR_BAD_VALUE. |
<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-18383751591</td>
<td>28780635</td>
<td>(Patch Set 2) There was an issue of wrong selection of item type when dealing with in-advance billing events. <code>/item/adjustment</code> was created instead of <code>/item/mcycle_forward_arrear</code> for a <code>cycle_forward_arrear</code> event when billing in-advance was set. This has been fixed.</td>
</tr>
<tr>
<td>3-16906450911</td>
<td>28786829</td>
<td>(Patch Set 2) The PCM_OP_BILL_MAKE_BILL_NOW opcode was resetting the error buffer set by a policy customization in the workflow. This has been fixed.</td>
</tr>
<tr>
<td>3-17857064951</td>
<td>28794484</td>
<td>(Patch Set 2) In the subscription module, if END_T passed was earlier than the <code>cycle/purchase/update</code> end dates, then the cycle start and end dates were set incorrectly leading to validation error. This has been fixed.</td>
</tr>
<tr>
<td>3-18482634741</td>
<td>28812794</td>
<td>(Patch Set 2) The Conversion Manager failed to load data into uniqueness table due to the presence of double quotes. This has been fixed to parse and omit double quotes in the uniqueness-related functions.</td>
</tr>
<tr>
<td>3-18482428661</td>
<td>28812798</td>
<td>(Patch Set 2) The PCM_OP_AR_BILL_ADJUSTMENT opcode had memory leaks. This has been fixed.</td>
</tr>
<tr>
<td>3-18310735251</td>
<td>28824835</td>
<td>(Patch Set 2) The Connection Manager module had some memory leaks. This has been fixed.</td>
</tr>
<tr>
<td>3-18505486831</td>
<td>28837038</td>
<td>(Patch Set 2) In the subscription module, for a backdated purchase done on the billing day of the month, the end date for the event was incorrectly set and this was causing wrong validity periods for the resources both in BRM and Pipeline Manager. This has been fixed.</td>
</tr>
<tr>
<td>3-17857944451</td>
<td>28843066</td>
<td>(Patch Set 2) The Load Price List utility was failing with core dump while committing or retrieving price list. This has been fixed.</td>
</tr>
<tr>
<td>3-16250214071</td>
<td>28848746</td>
<td>(Patch Set 2) The Billing Care user interface was not displaying the details of a bill that was generated before the account was moved into a hierarchy. This has been fixed.</td>
</tr>
<tr>
<td>3-18473654211</td>
<td>28848748</td>
<td>(Patch Set 2) The transaction opening workflow was not returning an error, while trying to open a transaction on a non-existent object. This has been fixed with proper checks for existence of an object before the object is tried for a lock to open the transaction on it.</td>
</tr>
<tr>
<td>3-18631761901</td>
<td>28871778</td>
<td>(Patch Set 2) The file name format of the General Ledger (G/L) reports generated in XML were different. This has been fixed to restore the earlier naming format.</td>
</tr>
<tr>
<td>3-18664098401</td>
<td>28873152</td>
<td>(Patch Set 2) Any auditupdate to the account creates an entry in the audit table for <code>au_account_t</code>. In addition, for a transaction that requires data from the audit table, searching in <code>au_account_t</code> was taking considerable time. In such cases, the updates had to fetch <code>effective_t</code> from <code>account_t</code> if <code>au_account_t</code> returned null. This has been fixed.</td>
</tr>
</tbody>
</table>
Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-18368278231</td>
<td>28905266</td>
<td>(Patch Set 2) When an account is deleted using a wrapper opcode that invokes PCM_OP_CUST_DELETE_ACCT opcode, an error in the workflow did not roll back the changes to uniqueness object. This has been fixed.</td>
</tr>
<tr>
<td>3-18263789131</td>
<td>28923379</td>
<td>(Patch Set 2) The PCM_OP_BILL_MAKE_BILL opcode finalized the bill when the workflow was invoked by a process other than <em>pin_bill_accts</em> indicated by the program name field. This has been fixed to finalize the bill only through the <em>pin_bill_accts</em> utility.</td>
</tr>
<tr>
<td>3-19353513291</td>
<td>28944041</td>
<td>(Patch Set 2) For PCI compliance, it was required to mask the PIN_FLD_SECURITY_ID field in the card attributes structure for enhanced card security. See &quot;Card Security Code Is Now Masked in Logs&quot;.</td>
</tr>
<tr>
<td>3-18447694071</td>
<td>28944043</td>
<td>(Patch Set 2) Paymentech DM has been enhanced to support Card Security Presence and to send Fraud Format Indicator (FR) record when this field is present. See &quot;Additional Card Security Presence Values Supported for Card Validation or Authorization&quot;.</td>
</tr>
<tr>
<td>3-18664468631</td>
<td>28944047</td>
<td>(Patch Set 2) Paymentech returned duplicate CIT and MIT transaction IDs. This has been fixed and transaction IDs for CIT and MIT records are generated in a random manner.</td>
</tr>
<tr>
<td>3-18713081091</td>
<td>28977386</td>
<td>(Patch Set 2) Oracle Data Manager was not responding for a long time when PIN_ERR_NO_MEM was returned from the flist copying method. This has been fixed.</td>
</tr>
<tr>
<td>3-18399785331</td>
<td>28997123</td>
<td>(Patch Set 2) In the subscription module, when multiple discounts were purchased with different validity dates in the same deal, the discounts rules were not set correctly for the individual discounts. The discount validity dates of the first discount was used for all other discounts in the deal. This has been fixed.</td>
</tr>
<tr>
<td>3-18310735251</td>
<td>29017375</td>
<td>(Patch Set 2) The Connection Manager was dumping core when running <em>pin_deferred_act</em> and <em>pin_bill_day</em> utilities due to memory management issues. This has been fixed.</td>
</tr>
<tr>
<td>3-18370747541</td>
<td>29023518</td>
<td>(Patch Set 2) Due to the scope of the header file, the customer management module was throwing a symbol look up error. This has been fixed.</td>
</tr>
<tr>
<td>3-18855799201</td>
<td>29036175</td>
<td>(Patch Set 2) In the subscription module, during the product cancellation operation, the refund amount was calculated incorrectly if it includes in-advance charges. This was due to considering event end date instead of cycle end date for calculating the scale for refund. This has been fixed.</td>
</tr>
<tr>
<td>3-18452678921</td>
<td>29042246</td>
<td>(Patch Set 2) During the billing delay period for a deployment with the billing delay configuration, any discount cancellation operation was not applying refunds. Only during the delay period this inconsistency was observed as the accounting dates available in the bill unit would not be updated and still reflect the previous cycle information. This has been fixed with proper checks to handle such events in the delay period.</td>
</tr>
</tbody>
</table>
### Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-18889790481</td>
<td>29042248</td>
<td>(Patch Set 2) A memory leak was observed in the customer registration module. This has been fixed.</td>
</tr>
<tr>
<td>3-18793585311</td>
<td>29042250</td>
<td>(Patch Set 2) In some cases, PCM_OP_EXEC_PROC was not returning the correct response. This has been fixed.</td>
</tr>
<tr>
<td>3-18919001443</td>
<td>29120662</td>
<td>(Patch Set 2) A memory leak was observed in the Paymentech DM. This has been fixed.</td>
</tr>
<tr>
<td>3-18919001133</td>
<td>29120695</td>
<td>(Patch Set 2) When creating /payinfo/cc objects, any custom fields passing into the workflow were not dropped, which caused a schema error. This has been fixed.</td>
</tr>
<tr>
<td>3-18693381421</td>
<td>29127140</td>
<td>(Patch Set 2) In the subscription module, the PCM_OP_SUBSCRIPTION_CANCEL_DEAL opcode caused segmentation fault core dump while processing discounts. This has been fixed.</td>
</tr>
<tr>
<td>3-16749719711</td>
<td>29141791</td>
<td>(Patch Set 2) When rerating was done before billing, the invoice data received from ECE was not present in the rerated event. Instead, original invoice data was populated in the rerated event. This has been fixed.</td>
</tr>
<tr>
<td>3-18964843081</td>
<td>29146203</td>
<td>(Patch Set 2) The modify customer operation was not applying the cycle discount information passed in the input due to missing Portal object ID (POID) information in PIN_FLD_DEAL_INFO. This has been fixed.</td>
</tr>
<tr>
<td>3-18125409791</td>
<td>29151194</td>
<td>(Patch Set 2) In the subscription module, canceling a purchased product with the cycle start date in the future and the purchase and usage start dates in the past, updated the cycle, purchase, and usage start dates. This has been fixed to update only the cycle start date.</td>
</tr>
<tr>
<td>3-1857842961</td>
<td>29151195</td>
<td>(Patch Set 2) For a purchased product, if either purchase, usage, or cycle end date was changed, the end date for purchase, usage, and cycle were updated. This has been fixed to update only the date that is changed.</td>
</tr>
<tr>
<td>3-18980345811</td>
<td>29174549</td>
<td>(Patch Set 2) In Collections Manager, the PCM_OP_COLLECTIONS_GROUP_CREATE opcode had memory leaks. This has been fixed.</td>
</tr>
<tr>
<td>3-18644785641</td>
<td>29178997</td>
<td>(Patch Set 2) Event adjustment was not working as expected if the amount or percentage was not provided in the input. The expectation was to do 100% event adjustment if the amount or percentage value was not provided. This has been fixed.</td>
</tr>
<tr>
<td>3-15270580261</td>
<td>29204382</td>
<td>(Patch Set 2) In an Application Integration Architecture (AIA) deployed environment, during any transaction where the request is sent from BRM to ECE, if an error occurred in ECE, then AIA was not able to capture the error from BRM. This has been fixed.</td>
</tr>
</tbody>
</table>
Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-15122386111</td>
<td>29221143</td>
<td>(Patch Set 2) The discount validity rules were not set for the discounts starting or ending in a future cycle as the accounting dates were prone to change. This has been fixed to identify the discounts during billing that have start or end dates in the next billing cycle and set the validity rules appropriately.</td>
</tr>
<tr>
<td>3-18536623001</td>
<td>29265113</td>
<td>(Patch Set 2) In case of a discount sharing scenario, when multiple reauthorization requests are created, the reservation quantity was not updated and it stayed at sum of the quantity of the first authorization and last reauthorization. In this case, it was possible to reauthorize sessions indefinitely and certain limits were also not enforced. This was leading to revenue leakage. This has been fixed.</td>
</tr>
<tr>
<td>3-19157592761</td>
<td>29278528</td>
<td>(Patch Set 2) In the subscription module, the plan transition opcode resulted in segmentation fault dumping core. This has been fixed.</td>
</tr>
<tr>
<td>3-18741993071</td>
<td>29342746</td>
<td>(Patch Set 2) While rating with sponsored product, the credit limit check was done both on the sponsored product and the user product. Due to this, a credit limit breach was reported when the user product sub-balance was positive. This has been fixed by removing the credit limit check on the user product while rating with the sponsored product.</td>
</tr>
<tr>
<td>3-19360664961</td>
<td>29342749</td>
<td>(Patch Set 2) CPU usage for the Connection Manager module reached and stayed at 100%. This has been fixed.</td>
</tr>
<tr>
<td>3-19278553351</td>
<td>29370866</td>
<td>(Patch Set 2) In the subscription module, the PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO opcode failed for an account-level discount. This has been fixed.</td>
</tr>
<tr>
<td>3-18738247231</td>
<td>29376633</td>
<td>(Patch Set 2) In the Pipeline Manager Framework, a memory leak was reported while converting decimal to string. This has been fixed.</td>
</tr>
<tr>
<td>3-19425603631</td>
<td>29395724</td>
<td>(Patch Set 2) In a Service Lifecycle Management (SLM)-enabled deployment, when a deal was purchased with balance as zero, the service state was immediately set to recharge_only. This caused usage processing issues in ECE. This has been fixed. Now, status is not changed to recharge_only as soon as the purchase is completed.</td>
</tr>
<tr>
<td>3-19456180091</td>
<td>29409245</td>
<td>(Patch Set 2) In the subscription module, during the product cancellation operation, the validity end dates for non-currency resources of an in-advance cycle was wrongly set. This has been fixed.</td>
</tr>
<tr>
<td>3-19353513291</td>
<td>29428268</td>
<td>(Patch Set 2) During the pin_collect process for payments, an error indicating PIN_ERR_PARTIAL was thrown, especially with CIT and MIT transactions. This has been fixed.</td>
</tr>
<tr>
<td>3-17880290091</td>
<td>29441735</td>
<td>(Patch Set 2) During the billing process, few items were incorrectly filtered during final billing, and it was due to item pruning in cases of billing suppression, suspension, or resume billing. This has been fixed.</td>
</tr>
</tbody>
</table>
Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-18995075711</td>
<td>29441740</td>
<td>(Patch Set 2) When a back-dated account with a multi-month cycle was moved into a hierarchy, the child account's PIN_FLD_ACTG_NEXT_T did not align with that of the parent account. Changes have been made to ensure that the dates are aligned.</td>
</tr>
<tr>
<td>3-18731886071</td>
<td>29448150</td>
<td>(Patch Set 2) Event adjustment was not working properly for tax-only adjustments. This has been fixed.</td>
</tr>
<tr>
<td>3-19391743111</td>
<td>29448154</td>
<td>(Patch Set 2) The PCM_OP_COLLECTIONS_GET_AGENTS_ACTIONS output flist did not return the PIN_FLD_COUNT field. As a result, the custom logic failed. This has been fixed.</td>
</tr>
<tr>
<td>3-19100404921</td>
<td>29448156</td>
<td>(Patch Set 2) In the subscription module, the product cancellation operation did not report missing validity information. This has been fixed.</td>
</tr>
<tr>
<td>3-19498301501</td>
<td>29454752</td>
<td>(Patch Set 2) Non-CIT and MIT were failing in an attempt to access the unavailable data. This has been fixed with appropriate null checks and error handling.</td>
</tr>
<tr>
<td>3-18953049631</td>
<td>29460620</td>
<td>(Patch Set 2) In the subscription module, incorrect discount amount was calculated when a discount was applied on a back-dated cycle forward monthly charge. This has been fixed.</td>
</tr>
<tr>
<td>3-19555740471</td>
<td>29463336</td>
<td>(Patch Set 2) In the subscription module, the <code>pin_cycle_fees</code> utility was not selecting the tailor made products when run in the <code>Cancel</code> mode. This has been fixed.</td>
</tr>
<tr>
<td>3-19125464031</td>
<td>29480201</td>
<td>(Patch Set 2) The AAA Services Framework, PCM_OP_TCF_AAA_AUTHORIZE opcode, had a memory leak. This has been fixed.</td>
</tr>
<tr>
<td>3-19522425971</td>
<td>29496597</td>
<td>(Patch Set 2) When a discount was purchased in deferred mode, and it was canceled subsequently before the actual start of the discount, the cycle, purchase, and usage start dates were not updated to align with the end dates. This has been fixed.</td>
</tr>
<tr>
<td>3-19525755941</td>
<td>29514020</td>
<td>(Patch Set 2) During billing, price-overridden products were not considered properly for rating. This has been fixed.</td>
</tr>
<tr>
<td>3-19583668901</td>
<td>29514025</td>
<td>(Patch Set 2) For a discount-sharing group, when balances were concurrently updated using the multithreaded application (MTA) framework, the balance updates were not made correctly obtaining necessary lock to the balance group. This has been fixed.</td>
</tr>
<tr>
<td>3-19611604151</td>
<td>29526005</td>
<td>(Patch Set 2) It was not possible to run multiple Rated Event (RE) Loader processes in parallel. When multiple RE Loader processes were run in parallel, the row lock contentions rendered the processes non-responsive. This has been fixed.</td>
</tr>
</tbody>
</table>
### Table 1-1 (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-19579803191</td>
<td>29533564</td>
<td>(Patch Set 2) The PCM_OP_PYMT_COLLECT opcode was not working for transaction ID with a length more than 16 characters. This has been increased to 22 characters to be consistent with the Paymentech (fusa) specifications.</td>
</tr>
<tr>
<td>3-19579899441</td>
<td>29533569</td>
<td>(Patch Set 2) Wrong quantity for the event was picked up for rating. The event matching logic was considering the events based on the order in which they appeared. This has been changed to the longest match event and now that is considered first, overriding the order of events.</td>
</tr>
<tr>
<td>3-19586933771</td>
<td>29533572</td>
<td>(Patch Set 2) Tax percentage was not added even if the tax calculation was not using any third-party tax engines. This has been fixed. Changes are done to check <code>tax_pkg_type</code> for zero-tax amount cases.</td>
</tr>
<tr>
<td>3-19568382071</td>
<td>29557884</td>
<td>(Patch Set 2) In Oracle Communications Billing Care, it was not possible to track the user making any changes or updates in the News Feed section. This has been fixed. The external user data is persisted as <code>PIN_FLD_EXTERNAL_USER</code> in the <code>newsfeed</code> storable class.</td>
</tr>
<tr>
<td>3-17713284711</td>
<td>29557888</td>
<td>(Patch Set 2) For an offer, if an usage event was rated that had 0 (zero) charge and then the offer was changed and rerating was 1 (one), there was no adjustment event created nor a new usage event was created with 0 charge. This has been fixed. Now, a new event with 0 charge is created and the general ledger ID (G/L ID) corresponding to the offer is impacted.</td>
</tr>
<tr>
<td>3-19600571011</td>
<td>29584658</td>
<td>(Patch Set 2) If the price of an one-time charge product is changed post billing and rerating and corrective invoicing is done, the rerated charge was not reflecting in the invoice. This has been fixed.</td>
</tr>
<tr>
<td>3-19624443371</td>
<td>29590523</td>
<td>(Patch Set 2) The RE Loader was failing with the error “too many open files” and required an update to handle maximum number of files. This has been fixed.</td>
</tr>
<tr>
<td>3-19733153701</td>
<td>29596720</td>
<td>(Patch Set 2) The <code>pin_deferred_act</code> utility was generating an error message in certain cases while dropping array elements, such as <code>PIN_FLD_TELCO_FEATURES</code>. This has been fixed.</td>
</tr>
<tr>
<td>3-19136247751</td>
<td>29622932</td>
<td>(Patch Set 2) A bill was not including the charges during service activation from deferred actions on the bill date. A new business parameter, <code>BillDeferredActions</code>, has been introduced in the <code>Billing</code> instance of the <code>config/business_params</code> object. By default, the <code>BillDeferredActions</code> parameter is <code>disabled</code> and the charges are added to the next bill. If you set the <code>BillDeferredActions</code> parameter to <code>enabled</code>, the charges are included in the current month bill.</td>
</tr>
<tr>
<td>3-19793794931</td>
<td>29634985</td>
<td>(Patch Set 2) Even when a migration failed, the Conversion Manager was reporting it as success. This has been fixed.</td>
</tr>
</tbody>
</table>
Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
</table>
| 3-19682040123 | 29673013   | (Patch Set 2) It was not possible to synchronize the account data loaded into BRM with ECE by using the Conversion Manager. By default, the BRM data is published to ECE asynchronously, which could potentially lead to inconsistent data between BRM and ECE, if there are any synchronization failures. Also, it was not possible to migrate the hierarchical accounts into BRM if the parent and child accounts are in the same files and the parent accounts were not loaded first. This has been fixed. You can now perform the following:  
  * Loading the migrated account data in BRM into ECE synchronously (in real-time).  
  * Migrate parent and child accounts into BRM using the same input file.  |
<p>| 3-19495959241 | 29681830   | (Patch Set 2) In Billing Care, it took longer time to display the Account page. It was due to the PCM_OP_AR_GET_ACCT_ACTION_ITEMS opcode, delaying loading of accounts to the Account page. This has been fixed.                                                                                                                                                                                                                          |
| 3-19188771891 | 29714064   | (Patch Set 2) When a product was canceled, refund was not going through if the product was purchased on Day 28 through Day 31 of a month with 31-day billing enabled and the next billing date did not exist in the subsequent month. This has been fixed.                                                                                                                                                                                                                       |
| 3-19855701651 | 29795746   | (Patch Set 2) BRM triggered ECE_POST_COMMIT in ECE even when the transaction was aborted. This has been fixed.                                                                                                                                                                                                                                                                                                                                                               |
| 3-20125667971 | 29808545   | (Patch Set 2) Any operation that included the PROFILE_SUBSCRIBER_PREFS_T table was causing a full table scan. This has been fixed by introducing an index on the table.                                                                                                                                                                                                                                                                                     |
| 3-19875296431 | 29820123   | (Patch Set 2) For the pin_mass_refund utility, performance issues were observed. This has been fixed.                                                                                                                                                                                                                                                                                                                                                                      |
| 3-20042957751 | 29858023   | (Patch Set 2) In the plan transition flow, a new notification event /event/notification/plan/transition/post_deal_cancel was required to be sent soon after cancellation. This has been added.                                                                                                                                                                                                                                 |
| 3-19883527851 | 29867410   | (Patch Set 2) During the cancellation of a cycle-forward event, the subscription module did not fetch all the related events for reversal of charges. This has been fixed.                                                                                                                                                                                                                                           |
| 3-17856477511 | 29871639   | (Patch Set 2) In the discounting module of Pipeline Manager, after unloading existing configuration, a memory leak was observed while loading the new configuration, which was causing the growth of the used memory. This has been fixed.                                                                                                                                                                                                |
| 3-18860602351 | 29900322   | (Patch Set 2) In the adjustment workflows, the decimal rounding precision was different from the configured precision for usage events. This has been fixed.                                                                                                                                                                                                                                                      |</p>
<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-19682040123</td>
<td>29920007</td>
<td>(Patch Set 2) In Conversion Manager, data errors were reported while migrating balances. These errors were due to unreleased database connections from the previous tasks. This has been fixed.</td>
</tr>
<tr>
<td>3-20036936261</td>
<td>29937126</td>
<td>(Patch Set 2) During the billing process, it was observed that the next item POID list of the service object was corrupted. This has been fixed.</td>
</tr>
<tr>
<td>3-20178161421</td>
<td>29960514</td>
<td>(Patch Set 2) While running the <code>pin_del_closed_accts</code> utility, NO_MEM error occurred. In such cases, memory was not cleared. This has been fixed.</td>
</tr>
<tr>
<td>3-191062741</td>
<td>29966023</td>
<td>(Patch Set 2) There were memory management issues in the Oracle Data Manager when search operation was performed using the PCM_OP_BILL_ITEM_ADJUSTEMENT opcode. This was because the search returned a huge number of events crossing the high watermark for the Data Manager process. This search template has been fixed to retrieve the calculated sum from the database instead of retrieving huge number of records and calculating the sum.</td>
</tr>
<tr>
<td>3-19353513291</td>
<td>29966026</td>
<td>(Patch Set 2) The Paymentech DM was not supporting Zero Value Account Verification (ZVAV) for American Express cards. This has been fixed.</td>
</tr>
<tr>
<td>3-19820667561</td>
<td>30031382</td>
<td>(Patch Set 2) The RE Loader Daemon could not process certain files resulting in unprocessed files pending in the input directory. This has been fixed.</td>
</tr>
<tr>
<td>3-2015367841</td>
<td>30095369</td>
<td>(Patch Set 2) The RE Loader process witnessed an unaccounted growth in memory due to memory leak. This has been fixed.</td>
</tr>
<tr>
<td>3-20574457081</td>
<td>30156863</td>
<td>(Patch Set 2) The <code>create_analytics_procedures_UTF8.plb</code> is now included.</td>
</tr>
<tr>
<td>3-1818837141</td>
<td>28653606</td>
<td>(Patch Set 2) The Daylight Saving Time calculation was incorrect for Brazil. This has been fixed.</td>
</tr>
<tr>
<td>3-17847235401</td>
<td>28923384, 28975424</td>
<td>(Patch Set 2) The RE Loader process was not gracefully shutdown, when it was still processing files. This has been fixed. Now, the RE Loader continues to process the pending files and then shuts down. A new configuration entry, <code>batch.shutdown.interval</code>, is introduced in the <code>BRM_Home/apps/pin_rel/Infranet.properties</code> file to set the shutdown interval in seconds. By default, this entry is set to 2.</td>
</tr>
<tr>
<td>3-15289852771</td>
<td>26810954</td>
<td>(Patch Set 1) Web Services Manager was not fully compliant to SOAP 1.2 format. This has been fixed.</td>
</tr>
<tr>
<td>3-15224841731</td>
<td>26810956, 27241793</td>
<td>(Patch Set 1) When the tiered taxation was applied, the tax information was consolidated instead of keeping it separate for each tier. This has been fixed.</td>
</tr>
</tbody>
</table>
Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-15019554421</td>
<td>26913721</td>
<td>(Patch Set 1) After a service transfer from one account to another using the PCM_OP_SUBSCRIPTION_SERVICE_BALGRP_TRANSFER opcode, the UNIQUENESS_T table was not updated with the correct account POID causing a mismatch with the correct account POID in the SERVICE_T table. This has been fixed.</td>
</tr>
<tr>
<td>3-15423599201</td>
<td>26954698</td>
<td>(Patch Set 1) For a quarterly billing cycle, the accounting dates were calculated incorrectly and this resulted in incorrect accounting periods. This has been fixed.</td>
</tr>
<tr>
<td>3-14874018361</td>
<td>26961858</td>
<td>(Patch Set 1) Refunding a fixed discount was incorrectly causing the refund of extra charges or other charges. This has been fixed.</td>
</tr>
<tr>
<td>3-15698161021</td>
<td>26968649</td>
<td>(Patch Set 1) Conversion Manager was throwing an error when the input XML contains a comma (,) in it. This has been fixed.</td>
</tr>
<tr>
<td>3-15795611881</td>
<td>26980059</td>
<td>(Patch Set 1) During product cancellation, while setting the product's charged period dates based on proration settings, the in-advance bill period was also being considered and this was causing inconsistencies. This has been fixed.</td>
</tr>
<tr>
<td>3-15360658251</td>
<td>26990857</td>
<td>(Patch Set 1) Authorization was incorrectly granting more than the available shared resource. This has been fixed.</td>
</tr>
<tr>
<td>3-15704767301</td>
<td>27001622</td>
<td>(Patch Set 1) In Collections Manager, when a scenario is replaced with another, the due date calculated for collection actions was incorrect. This has been fixed.</td>
</tr>
<tr>
<td>3-15224336401</td>
<td>27001628</td>
<td>(Patch Set 1) In Collections Manager, there was a deadlock situation during the collections process. This has been fixed.</td>
</tr>
<tr>
<td>3-15833386211</td>
<td>27001631</td>
<td>(Patch Set 1) When getting the aggregated information about a resource from BRM, the process was failing if the events for those resources were purged in the system. This has been fixed.</td>
</tr>
<tr>
<td>3-15122787861</td>
<td>27016775</td>
<td>(Patch Set 1) After the backdated modification of the discount end time (END_T), the charges were calculated incorrectly if sequential discounting was enabled. This has been fixed.</td>
</tr>
<tr>
<td>3-15504857121</td>
<td>27023170</td>
<td>(Patch Set 1) The PIN_ERR_LOG_FLIST function was generating core dump when any field in the flist logged was null. This has been fixed.</td>
</tr>
<tr>
<td>3-15377635591</td>
<td>27049618</td>
<td>(Patch Set 1) The 32-bit BRM client applications were not working properly with large files on Linux platforms. This has been fixed.</td>
</tr>
<tr>
<td>3-15805511541</td>
<td>27060891</td>
<td>(Patch Set 1) When the pin_collections_process utility was run, if the event POID was sent as null in the input flist while calling the PCM_OP_ARRESOURCE_AGGREGATION opcode, it was dumping core instead of returning a proper error message.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-1580938171</td>
<td>27067419</td>
<td>(Patch Set 1) When using Conversion Manager for loading data, the parent account and account receivable details could not be provided at bill-unit level. This was causing issues if the account had multiple bill units. This has been fixed.</td>
</tr>
<tr>
<td>3-16014529631</td>
<td>27067422</td>
<td>(Patch Set 1) The discount purchase was failing when the discount validity was set to full or none. This has been fixed.</td>
</tr>
<tr>
<td>NA</td>
<td>27116849</td>
<td>(Patch Set 1) The tax percentages were not properly getting accumulated in the balance impact of the rated event. This has been fixed.</td>
</tr>
<tr>
<td>3-15833122031</td>
<td>27130470</td>
<td>(Patch Set 1) There was an issue with state enumerations that was causing a core failure in BRM. This has been fixed.</td>
</tr>
<tr>
<td>3-15600725291</td>
<td>27159336</td>
<td>(Patch Set 1) The audit data retrieved from the database was not in the correct order. As a result, the rerating process was not creating charging events. This has been fixed.</td>
</tr>
<tr>
<td>3-16022883521</td>
<td>27159380</td>
<td>(Patch Set 1) When setting the discount validity dates on cycle boundaries, the discount validity rules were not considered for calculation and this resulted in incorrect discounts. This has been fixed.</td>
</tr>
<tr>
<td>3-13305479731</td>
<td>27159381</td>
<td>(Patch Set 1) There was a need to reassign the range for custom fields. This has been fixed.</td>
</tr>
<tr>
<td>3-16016616727</td>
<td>27178506</td>
<td>(Patch Set 1) Rerating was failing for usage events created on the billing day. This has been fixed.</td>
</tr>
<tr>
<td>3-15943051481</td>
<td>27181445</td>
<td>(Patch Set 1) For hierarchical accounts, bill-level adjustments were not getting synchronized with ECE properly. This has been fixed.</td>
</tr>
<tr>
<td>3-16240915721</td>
<td>27199237</td>
<td>(Patch Set 1) When applying cycle discounts, full discounts could not be applied due to incorrect discount start dates set by the set_discountinfo opcode. This has been fixed.</td>
</tr>
<tr>
<td>3-1597537621</td>
<td>27202235</td>
<td>(Patch Set 1) When accounts had hierarchical bill units, account deletion was failing. This has been fixed.</td>
</tr>
<tr>
<td>3-15997352581</td>
<td>27213533</td>
<td>(Patch Set 1) When the fetch size reached the maximum limit, the pin_bill_accts utility reported a performance issue. This has been fixed.</td>
</tr>
<tr>
<td>3-16129441831</td>
<td>27219879</td>
<td>(Patch Set 1) The pin_recover utility in the -rfr mode was not updating the result to zero in the charge event. This has been fixed.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-11520912431</td>
<td>27223026</td>
<td>(Patch Set 1) The subscription transfer workflow was not allowing to use the custom logic to control the rerate requests through policy. This has been fixed.</td>
</tr>
<tr>
<td>3-15695482521</td>
<td>27226303</td>
<td>(Patch Set 1) In BRM, it was possible to allocate the same adjustment to a bill twice. This has been fixed.</td>
</tr>
<tr>
<td>3-14772175791</td>
<td>27229728</td>
<td>(Patch Set 1) While loading offer profiles into BRM, the &quot;Double Free Or Corruption&quot; error was displayed. This has been fixed.</td>
</tr>
<tr>
<td>3-16342460241</td>
<td>27236217</td>
<td>(Patch Set 1) Rated Event Loader Daemon did not process backlog files and went into sleep mode. This has been fixed.</td>
</tr>
<tr>
<td>3-15233536061</td>
<td>27244352</td>
<td>(Patch Set 1) There was no option to disable journal creation for zero (0) balance records. This has been fixed.</td>
</tr>
<tr>
<td>3-15767584081</td>
<td>27263078</td>
<td>(Patch Set 1) Duplicate events were created for the same period for cycle-forward products and the events did not have any balance impacts associated with them. This has been fixed.</td>
</tr>
<tr>
<td>3-16228876211</td>
<td>27266272</td>
<td>(Patch Set 1) When PDC was used for configuring pricing objects, there was some issue in the Item Type Selector configuration due to format mismatch. This has been fixed.</td>
</tr>
<tr>
<td>3-16419713011</td>
<td>27269943</td>
<td>(Patch Set 1) The host name was not validated properly during the BRM installation in silent mode. This has been fixed.</td>
</tr>
<tr>
<td>3-16013792861</td>
<td>27274074</td>
<td>(Patch Set 1) The discount purchase operation was not considering the appropriate rate plan resulting in incorrect calculation of the discount period. This has been fixed.</td>
</tr>
<tr>
<td>3-15922064081</td>
<td>27274076</td>
<td>(Patch Set 1) For time-based full discounts, BRM was incorrectly calculating the discount amount. This has been fixed.</td>
</tr>
<tr>
<td>3-16186268621</td>
<td>27277544</td>
<td>(Patch Set 1) When a service configured to have only discounts was terminated, an error was thrown. This has been fixed.</td>
</tr>
<tr>
<td>3-16306116151</td>
<td>27288519</td>
<td>(Patch Set 1) For accounts with multiple bill units, write off reversal was failing as the reversal items pointed to one bill unit. This has been fixed.</td>
</tr>
<tr>
<td>3-16386155841</td>
<td>27304029</td>
<td>(Patch Set 1) Even if the log level was set to ERROR, some debug messages were getting recorded in the log files. This has been fixed.</td>
</tr>
<tr>
<td>3-16117097141</td>
<td>27304030</td>
<td>(Patch Set 1) The value for the <code>StopBillClosedAccounts</code> business parameter was always getting set to 1 irrespective of the value specified. This has been fixed.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-16335738541</td>
<td>27309569</td>
<td>(Patch Set 1) The input data for the payment policy opcode, PCM_OP_PYMT_POL_PRE_REFUND, did not contain all the required information for policy processing. This has been fixed.</td>
</tr>
<tr>
<td>3-16344647651</td>
<td>27309573</td>
<td>(Patch Set 1) The <code>pin_deposit</code> utility was not collecting pre-authorized charges. This has been fixed.</td>
</tr>
<tr>
<td>3-16103454181</td>
<td>27337657</td>
<td>(Patch Set 1) When making adjustments, if the adjustment amount was more than the actual due amount, an inconsistent behavior was observed. This has been fixed.</td>
</tr>
<tr>
<td>3-16407795601</td>
<td>27362564</td>
<td>(Patch Set 1) In Customer Center, when the permission to access account receivables (/accounttool/araccess) was set to read-only, the customer service representative (CSR) was not able to view the adjustment details from the <code>Bill Details</code> or <code>Item Details</code> page. This has been fixed.</td>
</tr>
<tr>
<td>3-16508349331</td>
<td>27369059</td>
<td>(Patch Set 1) System discounts for cycle arrear events were not applied properly. This has been fixed.</td>
</tr>
<tr>
<td>3-15223564141</td>
<td>27389701</td>
<td>(Patch Set 1) Web Services Manager was returning the Cannot initialize error when triggering the opcodes. This has been fixed.</td>
</tr>
<tr>
<td>3-15882582121</td>
<td>3-15832032811</td>
<td>27415116</td>
</tr>
<tr>
<td>3-16323836501</td>
<td>27418546</td>
<td>(Patch Set 1) The PCM_OP_COLLECTIONS_CALC_AGING_BUCKETS opcode was not considering the latest or current hierarchy to calculate the amount due from aging buckets. This has been fixed.</td>
</tr>
<tr>
<td>3-16663905461</td>
<td>27425461</td>
<td>(Patch Set 1) When installing BRM in silent mode, the response file required the database system user password which was a potential security risk. This has been fixed.</td>
</tr>
<tr>
<td>3-16634162461</td>
<td>27448146</td>
<td>(Patch Set 1) When the billing delay was configured, auto-triggered billing from ECE was generating a final bill instead of a partial bill. This has been fixed.</td>
</tr>
<tr>
<td>3-16380399871</td>
<td>27463239</td>
<td>(Patch Set 1) Launching Suspense Center with Java WebStart had issues due to the version mismatch between the shipped version (4.5.3) and the version specified (4.4) in the <code>orasoft.jnlp</code> file. This has been fixed.</td>
</tr>
<tr>
<td>3-16174721771</td>
<td>27463244</td>
<td>(Patch Set 1) The General Ledger job was taking more time than expected and required a performance update. This has been fixed.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-16264990561</td>
<td>27507372</td>
<td>(Patch Set 1) When a product configured with the cycle arrear event was canceled with a backdated date, the refund was not applied properly. This has been fixed.</td>
</tr>
<tr>
<td>3-16547489391</td>
<td>27523584</td>
<td>(Patch Set 1) Rerating after purchasing a discount was not working as expected. This has been fixed.</td>
</tr>
<tr>
<td>3-16250465901</td>
<td>27523592</td>
<td>(Patch Set 1) When a child account was moved under a parent account, the usage charges were not included in the correct bill. This has been fixed.</td>
</tr>
<tr>
<td>3-16628324941</td>
<td>27539381</td>
<td>(Patch Set 1) In some cases, while disconnecting a service, the PCM_OPCUST_SET_STATUS opcode was entering a deadlock. This has been fixed.</td>
</tr>
<tr>
<td>3-15870945191</td>
<td>27539388</td>
<td>(Patch Set 1) If sequential discounting was enabled, the subscription discounting was getting applied on top of the system-level discount. This was causing errors in some scenarios. This has been fixed.</td>
</tr>
<tr>
<td>3-16424349221</td>
<td>27552474</td>
<td>(Patch Set 1) During rerating, additional tax balance impacts were created. As a result, the adjustment calculated was incorrect. This has been fixed.</td>
</tr>
<tr>
<td>3-15922729871</td>
<td>27584537</td>
<td>(Patch Set 1) Conversion Manager was not handling migration of accounts with quarterly bill cycles properly. This has been fixed.</td>
</tr>
<tr>
<td>3-1651320161</td>
<td>27591136</td>
<td>(Patch Set 1) When event adjustment was done after tax-only adjustment, the tax jurisdiction information was not updated properly in the balance impact of the event adjustment. This has been fixed.</td>
</tr>
<tr>
<td>3-15903641491</td>
<td>27613611</td>
<td>(Patch Set 1) Rated Event Loader (pin_rel) was exiting due to the out of memory error. This has been fixed.</td>
</tr>
<tr>
<td>3-16506227591</td>
<td>27613617</td>
<td>(Patch Set 1) During cancellation of a product configured with in-advance billing, incorrect START_T and END_T were set in the events. This has been fixed.</td>
</tr>
<tr>
<td>3-16932753911</td>
<td>27621461</td>
<td>(Patch Set 1) In Collections Manager, when an account had two bill units, a paying bill unit and a non-paying bill unit, the aging due was calculated incorrectly. This has been fixed.</td>
</tr>
<tr>
<td>3-13832582461</td>
<td>27667959</td>
<td>(Patch Set 1) While recovering the failed payment, duplicate transaction IDs were observed and this was due to running two applications to recover the same failed payment, one was pin_recover and the other was a custom application, in parallel. This has been fixed.</td>
</tr>
<tr>
<td>3-16984338771</td>
<td>27667961</td>
<td>(Patch Set 1) The General Ledger job was taking more time than expected and required a performance update. This has been fixed.</td>
</tr>
</tbody>
</table>

Table 1-1  (Cont.) Customer-Reported Fixes for BRM
Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-16818528141</td>
<td>27684532</td>
<td>(Patch Set 1) In Collections Manager, if the PCM_OP_COLLECTIONS_GET_ACTION_HISTORY opcode was triggered with a transaction having a database number different from that of the collection event user, the opcode was failing. This has been fixed.</td>
</tr>
<tr>
<td>3-15685481061</td>
<td>27691744</td>
<td>(Patch Set 1) For an account with multiple bill units, a payment reversal for a backdated account level payment was having effective date as the current system date instead of the back date. This has been fixed.</td>
</tr>
<tr>
<td>3-16969300921</td>
<td>27745481</td>
<td>(Patch Set 1) The Oracle Data Manager process was failing with dumping core when processing the Account Receivables workflow. This has been fixed.</td>
</tr>
<tr>
<td>3-17041510991</td>
<td>27745484</td>
<td>(Patch Set 1) BRM was not providing options to control or customize workflow over rerate requests which were created as part of subscription transfer operation. This has been fixed.</td>
</tr>
<tr>
<td>3-16813237871</td>
<td>27745489</td>
<td>(Patch Set 1) When the EventAdjustmentDuringCancellation flag was enabled, the flist data corruption issue occurred intermittently, which was dumping core. This has been fixed.</td>
</tr>
<tr>
<td>3-16449323311</td>
<td>27751025</td>
<td>(Patch Set 1) The PCM_OP_AR_RESOURCE_AGGREGATION opcode was returning incorrect aggregated amount. This has been fixed.</td>
</tr>
<tr>
<td>3-16825381141</td>
<td>27771916</td>
<td>(Patch Set 1) The cancel discount operation was not setting the validity dates properly for discounts for which the end dates were in previous cycles. This has been fixed.</td>
</tr>
<tr>
<td>3-16736341981</td>
<td>27795345</td>
<td>(Patch Set 1) If sequential discounting was enabled, the PCM_OP_SUBSCRIPTION_SET_DISCOUNTINFO opcode was not working properly for in-advance billing periods. This has been fixed.</td>
</tr>
<tr>
<td>3-17062080421</td>
<td>27816387</td>
<td>(Patch Set 1) Even after paying the bill amount, the bill amount from the last bill was displayed as balance brought forward. This has been fixed.</td>
</tr>
<tr>
<td>3-16926420921</td>
<td>27881741</td>
<td>(Patch Set 1) When an event was rerated twice, the tax jurisdiction was not handled properly for adjustment events. This has been fixed.</td>
</tr>
<tr>
<td>3-16672265011</td>
<td>27881743</td>
<td>(Patch Set 1) When the same product was purchased twice, canceling one of the product deleted the provisioning tag from both the products. This has been fixed.</td>
</tr>
<tr>
<td>3-17181472981</td>
<td>27881746</td>
<td>(Patch Set 1) For billing cycles which spans across multiple months, newsfeed was getting created for the /event/notification/service_item/make_bill event for each accounting cycle. This has been fixed.</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1677875601</td>
<td>27891324</td>
<td>(Patch Set 1) In a prepaid environment, when the subscriber's balance was lower than the required balance for a purchase and if a deal was purchased, BRM displayed the credit limit exceeded error. In addition, signal 11 was observed in Connection Manager. This has been fixed.</td>
</tr>
<tr>
<td>3-17204427041</td>
<td>27894048</td>
<td>(Patch Set 1) The PCM_OP_SUBSCRIPTION_TRANSITION_PLAN opcode was not canceling the deals from the existing plan while moving to the new plan. This has been fixed.</td>
</tr>
<tr>
<td>3-17010229176</td>
<td>27894050</td>
<td>(Patch Set 1) During item adjustment, the search operation got into a infinite loop. This has been fixed.</td>
</tr>
<tr>
<td>3-16428542811</td>
<td>27938633</td>
<td>(Patch Set 1) The dm_fusa front end was not receiving data after the connection manager established connection with it. This has been fixed.</td>
</tr>
<tr>
<td>3-17136830661</td>
<td>27938637</td>
<td>(Patch Set 1) It was not possible to locate the Secure Sockets Layer (SSL) client wallet file due to the use of back slash () in the resource path instead of a forward slash (/). This has been fixed.</td>
</tr>
<tr>
<td>3-16509283701</td>
<td>27938640</td>
<td>(Patch Set 1) The PCM_OP_BILL_GET_ITEM_EVENT_CHARGE_DISCOUNT opcode was not allowing to add or modify the event data in its workflow through the PCM_OP_BILL_POL_GET_EVENT_SPECIFICDETAILS policy opcode. This has been fixed.</td>
</tr>
<tr>
<td>3-17133778471</td>
<td>27956401</td>
<td>(Patch Set 1) In Collections Manager, the collection write-off was failing for the bill unit in collection. This has been fixed.</td>
</tr>
<tr>
<td>3-16882624611</td>
<td>27979521</td>
<td>(Patch Set 1) The PCM_OP_SEARCH base opcode was failing when the template included a nested select query with the grouping by clause. This has been fixed.</td>
</tr>
<tr>
<td>3-17025208951</td>
<td>27993358</td>
<td>(Patch Set 1) During billing, the grantor information in the rollover bucket was incorrectly set to null. This has been fixed.</td>
</tr>
<tr>
<td>3-17399412831</td>
<td>27993360</td>
<td>(Patch Set 1) In LDAP Manager, the entry update operation was failing. This has been fixed.</td>
</tr>
<tr>
<td>NA</td>
<td>28007850</td>
<td>(Patch Set 1) When rating an event, sponsored products were ignored. This has been fixed.</td>
</tr>
<tr>
<td>3-17087785401</td>
<td>28020470</td>
<td>(Patch Set 1) If a service had two products and when one product was canceled, the EFFECTIVE_T value of the service was also updated. This was not allowing the processing of usage events for the other product. This has been fixed.</td>
</tr>
<tr>
<td>3-17232910091</td>
<td>28089237</td>
<td>(Patch Set 1) Quarterly cycles were getting merged which caused issues in applying charges. This has been fixed.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-17433817794</td>
<td>28089239</td>
<td>(Patch Set 1) The date format in Web Services Manager could not be configured based on the Java's SimpleDateFormat specification. This has been fixed.</td>
</tr>
<tr>
<td>3-17578946441</td>
<td>28119224</td>
<td>(Patch Set 1) When in-advance billing was enabled, purchasing discount with a future date to become valid on ACTG_NEXT_T resulted in validation errors. This has been fixed.</td>
</tr>
<tr>
<td>3-17548004211</td>
<td>28123147</td>
<td>(Patch Set 1) Conversion Manager was not providing an option to capture bill_accounting_cycles_left and next_bill_t information for migration. This has been fixed.</td>
</tr>
<tr>
<td>3-17548989291</td>
<td>28141241</td>
<td>(Patch Set 1) The dm_ldap process was failing to start when SSL was enabled. This has been fixed.</td>
</tr>
<tr>
<td>3-17298474221</td>
<td>28141243</td>
<td>(Patch Set 1) After loading the rated events from Pipeline Manager, it was found that MOD_T of the corresponding items and journals were less than the CREATED_T. This has been fixed.</td>
</tr>
<tr>
<td>3-16478416701</td>
<td>28159376</td>
<td>(Patch Set 1) When an adjustment was applied in BRM, the revision number for the balance group was set incorrectly to zero causing data inconsistency between BRM and ECE. This has been fixed.</td>
</tr>
<tr>
<td>3-17558077599</td>
<td>28200056</td>
<td>(Patch Set 1) After purchasing the monthly cycle forward product, the newsfeed for the same did not include the bill unit information. This has been fixed.</td>
</tr>
<tr>
<td>3-17120015936</td>
<td>28220691</td>
<td>(Patch Set 1) When there were multiple bill adjustments, reallocation of the payment was done incorrectly. This has been fixed.</td>
</tr>
<tr>
<td>3-17331083621</td>
<td>28247287</td>
<td>(Patch Set 1) The bill generated using Bill Now was not including all items due for billing. This has been fixed.</td>
</tr>
<tr>
<td>3-17302144931</td>
<td>28255178</td>
<td>(Patch Set 1) Even if there was no amount left for adjustment, BRM was allowing adjustment up to the full adjustment amount. This has been fixed.</td>
</tr>
<tr>
<td>NA</td>
<td>28259108</td>
<td>(Patch Set 1) When an APN Map was reloaded using a semaphore, the rating functionality did not consider the new APN Map configuration. This has been fixed.</td>
</tr>
<tr>
<td>3-17558202851</td>
<td>28261961</td>
<td>(Patch Set 1) BRM was considering the regular monthly cycle forward event as backdated if the billing was run after the specified in-advance billing days in the same month. This has been fixed.</td>
</tr>
<tr>
<td>3-17600593901</td>
<td>28261967</td>
<td>(Patch Set 1) For cycle forward events, having discounts the newsfeed functionality was not working. This has been fixed.</td>
</tr>
</tbody>
</table>
### Table 1-1  (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-17306790271</td>
<td>28278327</td>
<td>(Patch Set 1) Setting purchase start dates for first usage products was not allowed by using the PCM_OP_SUBSCRIPTION_SET_PRODINFO opcode. This has been fixed.</td>
</tr>
<tr>
<td>3-7711865221</td>
<td>28281697</td>
<td>(Patch Set 1) When BRM was processing orders by using JCA Adapter, if the CM instance goes down, the JCA Adapter was unable to restore the transaction state. As a result, the subsequent orders could not be processed without restart. This has been fixed.</td>
</tr>
<tr>
<td>3-17586682391</td>
<td>28334999</td>
<td>(Patch Set 1) In case of sequential discounting, expired or ended discounts were also applied when new discounts were purchased. This has been fixed.</td>
</tr>
<tr>
<td>3-17681700281</td>
<td>28342603</td>
<td>(Patch Set 1) When an account was created, the ACCOUNT_NO value was not updated correctly in the UNIQUE_ACCT_NO_T table. This has been fixed.</td>
</tr>
<tr>
<td>3-17540344751</td>
<td>28342608</td>
<td>(Patch Set 1) The event adjustment for the product configured for deferred taxes was not updating the tax jurisdiction information in the balance impacts correctly. This has been fixed.</td>
</tr>
<tr>
<td>3-16857082511</td>
<td>28366495</td>
<td>(Patch Set 1) There were memory leaks observed in the PCM_OP_TCF_AAA_STOP_ACCOUNTING opcode workflow. This has been fixed.</td>
</tr>
<tr>
<td>3-17834273241</td>
<td>28366497</td>
<td>(Patch Set 1) There was a memory leaks observed in the op_tcf_aaa_search_session workflow. This has been fixed.</td>
</tr>
<tr>
<td>3-17223827112</td>
<td>28379547</td>
<td>(Patch Set 1) In a BRM system integrated with ECE, External Manager (EM) Gateway was causing errors which led to data inconsistency. This has been fixed.</td>
</tr>
<tr>
<td>NA</td>
<td>28379549</td>
<td>(Patch Set 1) When the updated pricing data was loaded using semaphore, the charges were calculated incorrectly. This has been fixed.</td>
</tr>
<tr>
<td>3-17307745081</td>
<td>28379551</td>
<td>(Patch Set 1) With in-advance billing, the scale was calculated incorrectly as zero for a valid purchased product having an end date set. This has been fixed.</td>
</tr>
<tr>
<td>3-17867436786</td>
<td>28415082</td>
<td>(Patch Set 1) The PCM_OP_SUBSCRIPTION_TRANSITION_PLAN opcode was not canceling the deals from the existing plan while moving to the new plan. This has been fixed.</td>
</tr>
<tr>
<td>3-16017296621</td>
<td>28463456</td>
<td>(Patch Set 1) The rerating flow was not updating the session_obj and rerated_obj information correctly. This has been fixed.</td>
</tr>
<tr>
<td>3-16159206021</td>
<td>28466613</td>
<td>(Patch Set 1) When doing a plan transition, it was not possible to override the resource balance granted or add any custom workflow. This has been fixed.</td>
</tr>
</tbody>
</table>
Table 1-1 (Cont.) Customer-Reported Fixes for BRM

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-15944105871</td>
<td>28359386 28359381</td>
<td>(Patch Set 1) In Collections Manager, if schedule object was either completed or canceled, it was effective only for collection group owners, but not for members. This has been fixed.</td>
</tr>
</tbody>
</table>

Customer-Reported Fixes in ECE

Table 1-2 lists the service request (SR) issues reported by external sources for Oracle Communications Elastic Charging Engine (ECE) and provides a brief description of the resolution. The SRs are grouped by the respective patch sets.

Table 1-2 Customer-Reported Fixes for ECE

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-17841764601</td>
<td>28499167</td>
<td>(Patch Set 2) If there were multiple threshold events, then the custom data map was incorrect. This has been fixed.</td>
</tr>
<tr>
<td>3-18939351241</td>
<td>29252596</td>
<td>(Patch Set 2) With the discount sharing enabled in the system, the customer eviction was not working properly. This has been fixed.</td>
</tr>
<tr>
<td>3-18288658041</td>
<td>29269723</td>
<td>(Patch Set 2) The values stored for session initiate request of a session were not available during the update request. This has been fixed.</td>
</tr>
<tr>
<td>3-19167777401</td>
<td>29297126</td>
<td>(Patch Set 2) The latency information within External Manager (EM) Gateway were displayed incorrectly in the log files. This has been fixed.</td>
</tr>
<tr>
<td>3-19329380651</td>
<td>29352741</td>
<td>(Patch Set 2) For a long running session with split scenario, there were rounding issues with different scaling on charging and discounting. This has been fixed.</td>
</tr>
<tr>
<td>3-19236589851</td>
<td>29352848</td>
<td>(Patch Set 2) There were threading issues that were generating errors in the usage or update processing flows. This has been fixed.</td>
</tr>
<tr>
<td>3-17993347921</td>
<td>29361881</td>
<td>(Patch Set 2) The update processing flow was throwing a &quot;no valid product&quot; exception. This has been fixed.</td>
</tr>
<tr>
<td>3-18754665321</td>
<td>29383215</td>
<td>(Patch Set 2) Modifications to subbalance validity was not handled properly. This has been fixed.</td>
</tr>
<tr>
<td>3-19466561861</td>
<td>29415796</td>
<td>(Patch Set 2) Diameter Gateway was losing precision while processing integers that were big. This has been fixed.</td>
</tr>
<tr>
<td>3-19252747761</td>
<td>29422243</td>
<td>(Patch Set 2) If there was any change in the impact category, such as the user moving from one country to another in the same zone, the daily bundle was granted even if it was already granted. This has been fixed.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-19494708661</td>
<td>29428394</td>
<td>(Patch Set 2) For usages of 1KB, the rounding was not proper due to precision and scaling issues. This has been fixed.</td>
</tr>
<tr>
<td>3-19502473911</td>
<td>29479716</td>
<td>(Patch Set 2) The threshold breach notification was not including all Public User Identities (PUI) from the service alias list. This has been fixed.</td>
</tr>
<tr>
<td>3-19680335191</td>
<td>29531402</td>
<td>(Patch Set 2) During the usage processing, reverse rating was not working correctly when multiple discounts were evaluated. This has been fixed.</td>
</tr>
<tr>
<td>3-19393467861</td>
<td>29543203</td>
<td>(Patch Set 2) During the usage processing, discount evaluation was incorrect when multiple discounts were involved. This has been fixed.</td>
</tr>
<tr>
<td>3-19539308471</td>
<td>29634128</td>
<td>(Patch Set 2) Data in ECE cache was lost due to the growth of memory with unnecessary replication of Subscribe-Notifications-Requests (SNR). This has been fixed.</td>
</tr>
<tr>
<td>3-19858922041</td>
<td>29646046</td>
<td>(Patch Set 2) When processing usage, if there were multiple threshold breaches in the same session, the aggregated information of those breaches were not populated properly in the notification. This has been fixed.</td>
</tr>
<tr>
<td>3-18524076311</td>
<td>29647134</td>
<td>(Patch Set 2) The closed user group (CUG) evaluation of a customer who was earlier in a CUG was incorrectly done. This has been fixed.</td>
</tr>
<tr>
<td>3-19791677601</td>
<td>29669600</td>
<td>(Patch Set 2) When processing update requests, the rating profile was corrupted. This has been fixed.</td>
</tr>
<tr>
<td>3-19746265661</td>
<td>29674445</td>
<td>(Patch Set 2) When service balance group transfer was done within the same account, the item portal object IDs (POID) list and the next item POID list were not synchronized properly. This has been fixed.</td>
</tr>
<tr>
<td>3-19887122081</td>
<td>29763898</td>
<td>(Patch Set 2) Diameter Gateway was expecting proxyable bit in the command flag in credit control requests and this was causing Diameter Gateway to respond with an error DIAMETER_INVALID_HDR_BITS (3008) when the command flags were simply 0x80 for the request. This has been fixed and now Diameter Gateway does not validate for the proxyable bit in the command flags when processing credit control requests.</td>
</tr>
<tr>
<td>3-19452478281</td>
<td>29764019</td>
<td>(Patch Set 2) When a balance impact was null for a rated event, the database was generating a unique constraint violation. This has been fixed by marking the rated event as invalid.</td>
</tr>
<tr>
<td>3-19775119121</td>
<td>29782159</td>
<td>(Patch Set 2) In an ECE deployment with beat configured, the reverse rating was not considering the splits. This has been fixed.</td>
</tr>
<tr>
<td>3-19793612471</td>
<td>29808711</td>
<td>(Patch Set 2) A balance update request was failing with an exception during Subscribe-Notifications-Request (SNR) generation. This has been fixed.</td>
</tr>
</tbody>
</table>
Table 1-2 (Cont.) Customer-Reported Fixes for ECE

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-20146263101</td>
<td>29823002</td>
<td>(Patch Set 2) During the rolling upgrade, the tax code configuration data updated through Java Management Extensions (JMX) was lost. This was because only the charging-setting.xml was updated, but the updates were not reflected in the appconfiguration cache. This has been fixed.</td>
</tr>
<tr>
<td>3-19982052381</td>
<td>29843327</td>
<td>(Patch Set 2) The DiameterGy extension was throwing an exception on a top-up request. This has been fixed by adding sufficient checks.</td>
</tr>
<tr>
<td>3-19955343271</td>
<td>29960659</td>
<td>(Patch Set 2) During usage processing, a post File Update Information (FUI) update was trying to reserve further units causing DIAMETER_CREDIT_LIMIT_REACHED status to be returned to the network. This has been fixed.</td>
</tr>
<tr>
<td>3-20243888461</td>
<td>30013017</td>
<td>(Patch Set 2) Any updates to Oracle Communications Billing and Revenue Management (BRM) such as Life Cycle notifications from ECE to BRM were stuck in the Java Message Service (JMS) queue. The BRM Gateway was not able to send them to BRM due to connection pool issues. The BRM connections were not released back to the pool leading to non-availability of connections for any further requests. This has been fixed.</td>
</tr>
<tr>
<td>3-19847029811</td>
<td>30175640</td>
<td>(Patch Set 2) During usage processing, both validity time and granted service units were derived from adjusted rateable usage metric (RUM) quantity instead of the range-end. This has been fixed.</td>
</tr>
<tr>
<td>3-19676268161</td>
<td>30177988</td>
<td>(Patch Set 2) During usage processing, threshold notification was sent out based on reservation amount instead of actual consumed amount. This has been fixed.</td>
</tr>
</tbody>
</table>

Customer-Reported Fixes in PDC

Table 1-3 lists the service request (SR) issues reported by external sources for Oracle Communications Pricing Design Center (PDC) and provides a brief description of the resolution. The SRs are grouped by the respective patch sets.

Table 1-3 Customer-Reported Fixes for PDC

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-18956825171</td>
<td>29199299</td>
<td>(Patch Set 2) If the startDate in the bundleProductOffering was immediate, then the timestamp in the deal may be off by -1 hour. This caused failures while loading the deal to BRM. This has been fixed.</td>
</tr>
<tr>
<td>3-19414305551</td>
<td>29526136</td>
<td>(Patch Set 2) In Pricing Center, it was possible to configure proration settings for each of the charges under a charge selector. This was missing in PDC. This has been fixed. Now, the Charge Selector Configuration menu is available with options for Proration Settings and Remove Charge Selector. To configure the proration details, click Proration Settings.</td>
</tr>
</tbody>
</table>
### Table 1-3  (Cont.) Customer-Reported Fixes for PDC

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-19763604911</td>
<td>29590589</td>
<td>(Patch Set 2) While using the function <code>balance[Temporary Resource]</code>, an error was displayed in the PDC user interface. This has been fixed.</td>
</tr>
<tr>
<td>3-18417583781</td>
<td>29635057</td>
<td>(Patch Set 2) There were multiple issues while migrating real-time rating engine (RRE) and batch rating engine (BRE) data of BRM to PDC. BRE used multiple delayed events for the same service. Multiple PDC discount models created the discount model used by the discount selector and issues related to Plan and Deal transitions. This has been fixed.</td>
</tr>
<tr>
<td>3-17967410701</td>
<td>29661212</td>
<td>(Patch Set 2) User was not able to delete a ratable usage metric (RUM) because it was referenced in the older versions of the pricing and config objects. This has been fixed. Now, it is possible to delete the older versions of the referenced objects also, along with the RUM.</td>
</tr>
<tr>
<td>3-19142667251</td>
<td>29825919</td>
<td>(Patch Set 2) Pipeline rating was going wrong for the non-currency resources. Earlier, PDC was transforming the balance element's numeric code as the resource name to the BRM system, which was not the correct implementation. This has been fixed in migration and transformation to transform the balance element's code as the resource name instead of numeric code. This fix is relevant to anyone using PDC migration and transformation.</td>
</tr>
<tr>
<td>3-20003825861</td>
<td>29867480</td>
<td>(Patch Set 2) While configuring roll over charge offers using the PDC user interface, errors were displayed. This has been fixed.</td>
</tr>
<tr>
<td>3-20522364271</td>
<td>30150356</td>
<td>(Patch Set 2) If a zone model is referenced in a pricing object, such as charge offer, and a new zone model is created, the user is unable to obsolete or delete any zone model either through the user interface or <code>ImportExportPricing</code> utility. This has been fixed.</td>
</tr>
<tr>
<td>3-10534989821</td>
<td>20881534</td>
<td>(Patch Set 1) When the <code>Save as</code> option was used to copy a component, it was not copying only the component instead it copied the internally referenced components along with the component. This has been fixed.</td>
</tr>
<tr>
<td>3-11060700391</td>
<td>21575971</td>
<td>(Patch Set 1) After importing setup components, there were no rows created in certain tables which was causing issues. This has been fixed. When you install BRM, its installation scripts populate the sample data, including setup and pricing components. When using PDC, this data should not be initiated in BRM, but it should be defined in PDC and published to the BRM database. To ensure that PDC and BRM are integrated properly, you must clean up the BRM sample data before you start using PDC. See the discussion about cleaning the sample data in PDC Installation Guide.</td>
</tr>
<tr>
<td>3-11937523761</td>
<td>22488798</td>
<td>(Patch Set 1) The <code>ImportExportPricing</code> utility with the <code>-d</code> (delete) parameter worked only for obsoleting a setup component and only if the component was not referenced by any other component. After the component was made obsolete, it could not be used by any other components. This has been fixed.</td>
</tr>
</tbody>
</table>
### Table 1-3  (Cont.) Customer-Reported Fixes for PDC

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-13611868831</td>
<td>25091591</td>
<td>(Patch Set 1) Import of metadata was failing with a constraint violation if PDC was integrated with ECE. Recreate the index without COMPRESS ADVANCE LOW to fix this issue.</td>
</tr>
<tr>
<td>3-14513897441</td>
<td>25932012</td>
<td>(Patch Set 1) SyncPDC was failing if <code>event realtimeDiscount</code> class existed in the BRM database. Delete this class from the BRM database to fix this issue.</td>
</tr>
<tr>
<td>3-16039294067</td>
<td>27040318</td>
<td>(Patch Set 1) Syncpdc was failing during the transformation of the event if the usage class length was more than 4K, the maximum size of the column. This has been fixed.</td>
</tr>
<tr>
<td>3-15079324011</td>
<td>27756978</td>
<td>(Patch Set 1) PDC Weblogic Server managed server was logging verbose Eclipse Link messages (up to the FINE level) in the Weblogic Server log (nohup.out). The FINE level was not appropriate for Production environments. This has been fixed.</td>
</tr>
<tr>
<td>3-17231759901</td>
<td>27840554</td>
<td>(Patch Set 1) The BRM Integration Pack installation was failing because of certain privilege issues. A SYSDBA user credential was required to proceed. This has been fixed. The PDC Installer now accepts any user with database privileges not restricting to the SYSDBA user in the System Database Credentials screen.</td>
</tr>
<tr>
<td>NA</td>
<td>27881855</td>
<td>(Patch Set 1) When a rateable usage metric (RUM) was disassociated from all services and events in the service-event map, importing the RUM into PDC caused NULL pointer exception in the real-time rating transform engine. This has been fixed.</td>
</tr>
<tr>
<td>3-17267227961</td>
<td>27925018</td>
<td>(Patch Set 1) For the same recurring charge product definition, the rate was generated as fixed amount in the case of pricing center but as scaled amount in PDC. PDC actually supports fixed amount for recurring charges. This has been fixed.</td>
</tr>
<tr>
<td>3-17126536791</td>
<td>27963192</td>
<td>(Patch Set 1) When SyncPDC was run after changing the length of a field in the storable class associated with a custom event in BRM, the change was not reflected in the event definition in the PDC database. This has been fixed.</td>
</tr>
<tr>
<td>3-17317338671</td>
<td>28094045</td>
<td>(Patch Set 1) The high-volume PDC tables were not getting purged which was leading to performance issues. This has been fixed.</td>
</tr>
<tr>
<td>3-16272784751</td>
<td>28171359</td>
<td>(Patch Set 1) For chargeshares, the General Ledger ID was migrated as Undefined instead of Not Set. This has been fixed.</td>
</tr>
<tr>
<td>3-17677726731</td>
<td>28262068</td>
<td>(Patch Set 1) The USC_MAP was case sensitive whereas the ZONE_RESULT was case insensitive. This has been fixed. The USC_MAP is also case insensitive.</td>
</tr>
</tbody>
</table>
Customer-Reported Fixes in Billing Care

Table 1-4 lists the service request (SR) issues reported by external sources for Oracle Communications Billing Care and provides a brief description of the resolution. The SRs are grouped by the respective patch sets.

Table 1-4  Customer-Reported Fixes for Billing Care

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-16951952221</td>
<td>27764510</td>
<td>(Patch Set 2) In the Allocate Adjustment and Payment Details dialog boxes, the columns were not resizable for simple and advanced views. This has been fixed and the columns are resizable now.</td>
</tr>
<tr>
<td>3-17556441791</td>
<td>28466526</td>
<td>(Patch Set 2) In News Feed, when the Show Payment in Suspense link was clicked for a suspended payment, a payment suspense search page opened in a new browser tab. This has been fixed and clicking the link now opens the suspended payment record.</td>
</tr>
<tr>
<td>3-17876252841</td>
<td>28543516</td>
<td>(Patch Set 2) Terminating or Inactivating discounts for a service was throwing an error. This has been fixed.</td>
</tr>
<tr>
<td>3-18501913282</td>
<td>28853673</td>
<td>(Patch Set 2) It was not possible to log in to Billing Care from a new browser tab using a bookmarked URL, if there was a browser tab already logged in with Billing Care. This has been fixed.</td>
</tr>
</tbody>
</table>
### Table 1-4  (Cont.) Customer-Reported Fixes for Billing Care

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-18508445081</td>
<td>28806805</td>
<td>(Patch Set 2) Some of the dependent classes for generating reports were not shipped as part of the REST WAR files package. This has been fixed and the missing dependencies are now included.</td>
</tr>
<tr>
<td>3-18298738421</td>
<td>28837041</td>
<td>(Patch Set 2) The connection to BRM was lost after timeout exception was thrown for the operations wrapped under a transaction. This has been fixed.</td>
</tr>
<tr>
<td>3-18506198641</td>
<td>28854728</td>
<td>(Patch Set 2) Transfers between billable credit and debit items were treated as accounts receivable. This resulted in incorrect breakup of charges under the Bills tab. This has been fixed.</td>
</tr>
<tr>
<td>3-18504936078</td>
<td>28905269</td>
<td>(Patch Set 2) While navigating to accounts receivable action details, after selecting a refund type accounts receivable action, the Back button was not working and its label was missing. This has been fixed.</td>
</tr>
<tr>
<td>3-18613272481</td>
<td>28905272</td>
<td>(Patch Set 2) It was not possible to install Billing Care REST in silent mode. This has been fixed.</td>
</tr>
<tr>
<td>3-18504936134</td>
<td>28912852</td>
<td>(Patch Set 2) Billing Care now displays a tooltip in the Bill ID field of the following dialog boxes: Payment Details, Payment Details Item Affected, Account Adjustment, and Account Adjustment Details.</td>
</tr>
<tr>
<td>3-18504936071</td>
<td>28912854</td>
<td>(Patch Set 2) The Cancel link was not working for the notes in the Event Adjustment Details dialog box. This has been fixed.</td>
</tr>
<tr>
<td>3-18801505311</td>
<td>29048170</td>
<td>(Patch Set 2) When switching between bills, there was an issue with the Bills tab Show/Hide link. This has been fixed.</td>
</tr>
<tr>
<td>3-18605545081</td>
<td>29161373</td>
<td>(Patch Set 2) The Purchase Package and Purchase Deal flows are now supported as embeddable URLs. See &quot;Embeddable URLs for Purchase Package and Purchase Bundle Screens&quot; for more information.</td>
</tr>
<tr>
<td>3-16250214071</td>
<td>29161376</td>
<td>(Patch Set 2) Billing Care is now enhanced to display bills that were generated before the account was moved under a hierarchy. See &quot;View Bills Generated Before Moving the Account to a Hierarchy&quot; for more information.</td>
</tr>
<tr>
<td>3-18979119339</td>
<td>29161377</td>
<td>(Patch Set 2) It was not possible to enter special characters for the Billing Care Wallet Password field. This has been fixed.</td>
</tr>
<tr>
<td>3-19040269941</td>
<td>29272365</td>
<td>(Patch Set 2) The Refund Bill dialog box was displaying an error that the amount was greater than 999.99. This has been fixed.</td>
</tr>
<tr>
<td>3-19112468071</td>
<td>29272368</td>
<td>(Patch Set 2) There were performance issues with the bill details view. It was taking more time to load particularly for the flows involving Billing Care URLs embedded in other applications. This has been fixed.</td>
</tr>
<tr>
<td>3-19133358468</td>
<td>29290979</td>
<td>(Patch Set 2) The collection actions scheduled dates were one day older than the actual due date. This has been fixed to show the correct scheduled dates.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-19120021761</td>
<td>29290981</td>
<td>(Patch Set 2) If an account is created using customer center with a country, for example USA, and the same account was opened in Billing Care, a different country was displayed in the account profile details. This has been fixed.</td>
</tr>
<tr>
<td>3-19076461581</td>
<td>29323429</td>
<td>(Patch Set 2) There were alignment issues with the labels of the Tax Setup dialog box under Financial Setup. This has been fixed.</td>
</tr>
<tr>
<td>3-18957602481</td>
<td>29349021</td>
<td>(Patch Set 2) Discounts which were overridden as part of product purchase were not displayed in the breakup of the Original Charge or Discount or Net details in the Events dialog box. This has been fixed.</td>
</tr>
<tr>
<td>3-19161875121</td>
<td>29376636</td>
<td>(Patch Set 2) It is now possible to display custom error messages on the Billing Care user interface when purchasing a bundle.</td>
</tr>
<tr>
<td>3-19152779561</td>
<td>29376640</td>
<td>(Patch Set 2) It was not possible to link purchase reason and notes to a particular add-on purchase event. This has been fixed. A new method has been introduced, which retrieves the list of all the purchase events that can be used in the Billing Care SDK to assign to the notes.</td>
</tr>
<tr>
<td>3-18937531041</td>
<td>29395718</td>
<td>(Patch Set 2) To improve the performance of the Assets user interface, lazy loading of the assets has been implemented. By default, limited service information is retrieved and when the customer service representatives expand the particular service card, the additional service and offer details are retrieved.</td>
</tr>
<tr>
<td>3-19317248768</td>
<td>29428269</td>
<td>(Patch Set 2) The <strong>Bill in progress</strong> text was displayed twice in the Events page header. This has been fixed.</td>
</tr>
<tr>
<td>3-19374947041</td>
<td>29435523</td>
<td>(Patch Set 2) After installation, Business Operation Center was opening with a blank screen. This has been fixed.</td>
</tr>
<tr>
<td>3-19172328801</td>
<td>29441732</td>
<td>(Patch Set 2) Overdue days were shown incorrectly as the last configured period of the aging buckets. This has been fixed to show the actual overdue days. A new popup has been added to list the age of the overdue balances as per the aging bucket configuration.</td>
</tr>
<tr>
<td>3-19375982111</td>
<td>29441738</td>
<td>(Patch Set 2) Reason code was missing in the make payment flow. This has been fixed and the reason code is added now.</td>
</tr>
<tr>
<td>3-19549400461</td>
<td>29493386</td>
<td>(Patch Set 2) The Event Adjustment dialog box can now be customized to separate both <strong>Adjust the Amount</strong> field and <strong>Adjust the Percent</strong> field. Also, there was an issue with tax only adjustment. This has been fixed.</td>
</tr>
<tr>
<td>3-18127339011</td>
<td>29511807</td>
<td>(Patch Set 2) There was a performance issue while retrieving users who had access to the <strong>Payment Suspense</strong> screen because all the configured users were queried. This has been fixed and it is now possible to query only users associated with a specific role who has the required screen access.</td>
</tr>
<tr>
<td>3-18930292991</td>
<td>29520395</td>
<td>(Patch Set 2) The Billing Care SDK is now enhanced to support implementing the logic to restrict the deal list based on the role of the customer service representative. See <strong>Billing Care SDK Guide</strong>.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-19083734451</td>
<td>29629576</td>
<td>(Patch Set 2) The Billing Care SDK is now enhanced to support implementing the logic for filtering deals list to display manual discount deals only. See <em>Billing Care SDK Guide</em>.</td>
</tr>
<tr>
<td>3-19040270003</td>
<td>29629577</td>
<td>(Patch Set 2) Object serialization error was logged in the Oracle Entitlements Server client weblogic logs. This has been fixed.</td>
</tr>
<tr>
<td>3-19317720898</td>
<td>29645139</td>
<td>(Patch Set 2) The future dated add on purchase were purchased with wrong dates. This has been fixed.</td>
</tr>
<tr>
<td>3-18784241191</td>
<td>29649643</td>
<td>(Patch Set 2) Billing Care is now enhanced to display the external user names (login user) in News Feed to clearly highlight who has made changes on the account or have performed any account receivables actions. See <em>Billing Care SDK Guide</em>.</td>
</tr>
<tr>
<td>3-19083920361</td>
<td>29705003</td>
<td>(Patch Set 2) It is now possible to customize Billing Care to include start date and end date based filtering for add on deal purchase.</td>
</tr>
<tr>
<td>3-19533307321</td>
<td>29759310</td>
<td>(Patch Set 2) In the Billing Care Item Adjustment dialog box, it is possible to see the total amount for which adjustment is done. But it does not display how much other amount was adjusted against specific item. This has been fixed.</td>
</tr>
<tr>
<td>3-19530710241</td>
<td>29759312</td>
<td>(Patch Set 2) To improve performance, Billing Care has been enhanced to disable direct click on the Assets view from the Home tab through customization. It is now possible to search through service ID to go to the Assets page which display only services associated with the search criteria.</td>
</tr>
<tr>
<td>3-19549400461</td>
<td>29789695</td>
<td>(Patch Set 2) The Oracle Entitlements Server obligation check was throwing an exception. A null value check on adjustment amount when percentage only adjustment is done using customization was not introduced to avoid this exception. This has been fixed.</td>
</tr>
<tr>
<td>3-16426973211</td>
<td>27375507</td>
<td>(Patch Set 1) In Billing Care, it was not possible to save the notes in the Payment Details dialog box when the amount was more than 999 and when the payment was added by using opcode. This has been fixed.</td>
</tr>
<tr>
<td>3-16355498921</td>
<td>27423362</td>
<td>(Patch Set 1) When an attempt was made to open details using the account number hyperlink in the Payment Suspense Page Audit trail, it was opening the incorrect Suspense Account page. This has been fixed.</td>
</tr>
<tr>
<td>3-16726467291</td>
<td>27500263</td>
<td>(Patch Set 1) When a payment was made and then reversed, if a comment was added under reversal section, it was not visible unless the Payment Details dialog box was reopened. This has been fixed.</td>
</tr>
<tr>
<td>3-16982060751</td>
<td>27698947</td>
<td>(Patch Set 1) In Billing Care, a performance issue was reported while listing the deals during a product purchase operation. This has been fixed.</td>
</tr>
<tr>
<td>3-15485573861</td>
<td>27722599</td>
<td>(Patch Set 1) In Billing Care, in the Payment Details dialog box, the payment allocation table was not aligned properly. This has been fixed.</td>
</tr>
<tr>
<td>SR Number</td>
<td>Bug Number</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3-16918562891</td>
<td>27771469</td>
<td>(Patch Set 1) If a bill contains charges from subordinate bill units, the items were getting over allocated. This has been fixed.</td>
</tr>
<tr>
<td>3-17110341365</td>
<td>27823059</td>
<td>(Patch Set 1) The first column was <strong>Name</strong> in the Allocate Adjustment screen instead of <strong>Account</strong>. This has been fixed.</td>
</tr>
<tr>
<td>3-17102959567</td>
<td>27828889</td>
<td>(Patch Set 1) The Payment Allocation screen displayed the account POID instead of the account number. This has been fixed.</td>
</tr>
<tr>
<td>3-17069559841</td>
<td>27874943</td>
<td>(Patch Set 1) When a payment was first applied to an account and then moved to suspense and then again allocated to another account and then moved to suspense, the Audit Trail was not updated correctly. It was still showing the first account. This has been fixed.</td>
</tr>
<tr>
<td>3-16991351901</td>
<td>27874945</td>
<td>(Patch Set 1) While purchasing product/discount, the dates were not displaying correctly if there were timezone differences. This has been fixed.</td>
</tr>
<tr>
<td>3-16770126831</td>
<td>27924915</td>
<td>(Patch Set 1) Using the Billing Care SDK, it was not possible to customize the purchase deal screen and add a few custom fields. Also it was not possible to display the new fields in the asset display screen, changing the start/end date and change the overridden amount. This has been fixed. See <strong>Billing Care SDK Guide</strong>.</td>
</tr>
<tr>
<td>3-17270002631</td>
<td>27931964</td>
<td>(Patch Set 1) If a bundle was purchased with an override amount, the override amount was not accepted if it included &quot;,&quot;. This has been fixed.</td>
</tr>
<tr>
<td>3-17304458501</td>
<td>27938642</td>
<td>(Patch Set 1) The date in the Payment Reversal page after posting a comment was not displayed correctly unless the page was refreshed. Also, the first comment was not visible. This has been fixed.</td>
</tr>
<tr>
<td>3-17515609821</td>
<td>28206783</td>
<td>(Patch Set 1) Whenever a customization was added to Billing Care, the cache was not getting refreshed automatically. The user had to clear the browser cache and relogin. This has been fixed.</td>
</tr>
<tr>
<td>3-17402905141</td>
<td>28220688</td>
<td>(Patch Set 1) The override amount was not being retained when the product status was changed to active. This has been fixed.</td>
</tr>
<tr>
<td>3-17346452171</td>
<td>28261963</td>
<td>(Patch Set 1) The search results in the Payment Suspense screen was not displaying all the records as per the set pagination limit. This has been fixed.</td>
</tr>
<tr>
<td>3-17876082331</td>
<td>28431427</td>
<td>(Patch Set 1) The Account Details page for an account with multiple bill units was taking more time to open which was leading to time out. This has been fixed.</td>
</tr>
</tbody>
</table>
### Table 1-4  (Cont.) Customer-Reported Fixes for Billing Care

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-1785661511</td>
<td>28543514</td>
<td>(Patch Set 1) When a credit adjustment was applied to a bill with no dues, the Event Adjustment column in the adjust event actions screen was showing only the tax part of the event adjustment instead of the full adjustment amount. This has been fixed.</td>
</tr>
<tr>
<td>3-1749789341</td>
<td>28543965</td>
<td>(Patch Set 1) The DataModelGenerator utility was not generating custom XSD for custom objects available in BRM. This has been fixed.</td>
</tr>
<tr>
<td>3-178475681</td>
<td>28543979</td>
<td>(Patch Set 1) When Billing Care submitted the batch payments through Lockbox file, the entire file was rolled back if it included under payments. This has been fixed.</td>
</tr>
<tr>
<td>3-17943001841</td>
<td>28543981</td>
<td>(Patch Set 1) Account details were not being displayed in the Billing Care Collections Page after implementing custom collections scenario and customizing the policy opcode. This has been fixed.</td>
</tr>
</tbody>
</table>

### Customer-Reported Fixes in Business Operations Center

**Table 1-5** lists the service request (SR) issues reported by external sources for Oracle Communications Business Operations Center and provides a brief description of the resolution. The SRs are grouped by the respective patch sets.

### Table 1-5  Customer-Reported Fixes for Business Operations Center

<table>
<thead>
<tr>
<th>SR Number</th>
<th>Bug Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-18689185011</td>
<td>28602921</td>
<td>(Patch Set 2) The SERVICE_HOST parameter in Business Operations Center configuration properties file restricted login with IP address when the parameter was set as hostname, and restricted login with hostname when the parameter was set as IP address. This has been fixed. Dependency on the SERVICE_HOST parameter has been removed to construct REST URL.</td>
</tr>
<tr>
<td>3-18758155401</td>
<td>29033987</td>
<td>(Patch Set 2) The Business Operations Center silent installer has been enhanced to update wallet with Secure Shell user and BRM connection keys.</td>
</tr>
<tr>
<td>3-19190623931</td>
<td>29290982</td>
<td>(Patch Set 2) During validation, the Business Operations Center installer was not accepting special character for passwords in the Wallet field. This has been fixed. Now, special characters in addition to numbers and characters are accepted.</td>
</tr>
<tr>
<td>3-19558733471</td>
<td>29699403</td>
<td>(Patch Set 2) The Business Operations Center failure report was showing Account Portal object ID (POID) instead of the account number. This has been fixed and the report now shows the account number.</td>
</tr>
</tbody>
</table>
New Features in BRM

This chapter provides an overview of the new features in Oracle Communications Billing and Revenue Management (BRM) 12.0 Patch Set 1 and Patch Set 2.

Topics in this document:

• New Features in BRM 12.0 Patch Set 2
• New Features in BRM 12.0 Patch Set 1

New Features in BRM 12.0 Patch Set 2

BRM 12.0 Patch Set 2 includes the following enhancements:

• Deploying BRM Services on a Cloud Native Environment
• Wholesale Billing Enhancements
• Additional Card Security Presence Values Supported for Card Validation or Authorization
• Card Security Code Is Now Masked in Logs
• Display Bills Generated Before Moving the Account to a Hierarchy
• Managing Discount Validity Starting or Ending in Future Cycles
• Wildcard in Item Type Selectors
• BRM Supports POID Generation in ECE
• Support for Stored-Credential Transactions for Payments
• Support for Migration of Legacy Data into BRM and ECE in Real Time
• Support for Migrating Hierarchical Accounts Using Same Input File
• Event Rounding Rules Can Be Used for Adjustments
• New Staging Directory for Exported G/L reports
• BRM Client Applications Supported from 12.0 Patch Set 2

Deploying BRM Services on a Cloud Native Environment

BRM, Oracle Communications Pricing Design Center (PDC), Billing Care, and Business Operations Center now support their deployment on a cloud native environment, allowing you to harness the benefits of cloud with the services of BRM.

To deploy these BRM services, you use the new Oracle Communications BRM Cloud Native Deployment Option. This automates the deployment of BRM products and speeds up the process to get services up and running. Product deployments are preconfigured to communicate with each other through Helm charts.

For more information, see BRM Cloud Native Installation and Administration Guide.
Wholesale Billing Enhancements

With this enhancement, the following are supported in wholesale billing:

- The usage charges calculated by ECE and Pipeline Manager are also considered for wholesale billing.

During billing, all the usage charges applied to the nonpaying child bill units (wholesale child accounts) are aggregated and rolled up to the paying parent bill unit (wholesale parent account). When rated events of the nonpaying child accounts are loaded into BRM, the /tmp_journals_to_process objects are created in BRM instead of the /journal objects. The /tmp_journals_to_process objects are created if deferred taxation is configured and the CycleTaxInterval business parameter is set to billing. See “Specifying How to Calculate Deferred Taxes for Wholesale Billing” for setting the CycleTaxInterval business parameter and the discussions about enabling and disabling taxation globally in BRM Calculating Taxes for configuring deferred taxation.

- You can move child bill units into a wholesale hierarchy or move them out of a wholesale hierarchy even if there are pending charges or due amounts. The due amount of the child bill unit or the bill generated before moving the child bill unit is moved along with the child bill unit. You can generate invoice and apply the payment for that due amount after moving the child bill unit.

- You can generate invoices for both paying parent bill unit and nonpaying child bill units.

In the Summary of Current Charges section in the parent bill unit's invoice, you can view the sum of the service-level charges, taxes, and surcharges rolled up from the nonpaying child bill units as account-level charges. The adjustment details for the child bill units are not displayed in the parent bill unit's invoice as they are performed at account-level. However, the adjustments in the parent bill unit's invoice includes the sum of all the adjustments performed for the nonpaying child bill units.

In the child bill unit's invoice, you can view the plan and account receivable (A/R) action details for that specific bill unit.

- The /tmp_ar_item_to_roll_up objects are enabled for partitioning automatically.

You can create the /tmp_ar_item_to_roll_up objects for wholesale billing in partitions and purge these objects from the database by using the partition_utils utility. When the partition_utils utility is run, the /tmp_ar_item_to_roll_up objects with the status as 1 (processed) are purged. You can run the partition_utils utility for the specific time interval.

Specifying How to Calculate Deferred Taxes for Wholesale Billing

You can specify how BRM calculates deferred taxes for wholesale billing by setting the CycleTaxInterval business parameter to billing.

When set to billing, the tax is forwarded from the child account to the parent account. BRM calculates taxes for the parent account only, but the single tax item on the parent account includes taxes from both the parent and child accounts.

To specify how to calculate deferred taxes for wholesale billing:
1. Go to the `BRM_Home/sys/data/config` directory, where `BRM_Home` is the directory where you installed BRM components.

2. Create an XML file from the `config/business_params` object:
   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
   This command creates the XML file named `bus_params_billing.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_billing.xml.out` file.

4. Search for the following line:
   ```
   <CycleTaxInterval>accounting</CycleTaxInterval>
   ```

5. Change `accounting` to `billing`.

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

6. Save this file as `bus_params_billing.xml`.

7. Load the XML file into the BRM database:
   ```
   pin_bus_params bus_params_billing.xml
   ```

8. Stop and restart Connection Manager (CM).

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

**Additional Card Security Presence Values Supported for Card Validation or Authorization**

For credit card validations or authorizations, Paymentech Data Manager (`dm_fusa`) sends the Fraud Format Indicator (FR) or Product Record (PFR) record with the card security presence value. When the card security code (such as VISA CVV2) is present in the PIN_FLD_SECURITY input flist field, Paymentech DM sets the card security presence value to 1 and sends the FR or PFR record for validation by default.

With this enhancement, you can customize BRM opcodes to send other card security presence values (such as 9 (No Value)) in the input flist to Paymentech DM. You can use the PIN_FLDAVAILABLE flist field to provide the card security presence value.

   **Note:**
   The PIN_FLDAVAILABLE field can be used for both online and batch transactions.
Following are the enumerated names and values supported in the PIN_FLD_AVAILABLE field:

```c
typedef enum pin_pymt_card_secid_presence {
  PIN_PYMT_CSP_BLANK = 0,
  PIN_PYMT_CSP_AVAILABLE = 1,
  PIN_PYMT_CSP_ILLEGIBLE = 2,
  PIN_PYMT_CSP_NOT_PROVIDED = 5,
  PIN_PYMT_CSP_NO_CSV = 9,
} pin_pymt_card_secid_presence_t;
```

These names and values are defined in the `BRM_home/include/pin_pymt.h` file.

You can customize the `PCM_OP_PYMT_POL_PRE_COLLECT` opcode to send PIN_FLD_AVAILABLE and PIN_FLD_SECURITY_ID in the input flist.

For credit card transactions, Paymentech Data Manager now does the following:

- If only PIN_FLD_SECURITY_ID value is present in the input flist, Paymentech DM sends the FR or PFR record for validation with the card security presence value set to 1 (value present).
- If both PIN_FLD_SECURITY_ID and PIN_FLD_AVAILABLE values are present in the input flist, Paymentech DM sends the FR or PFR record for validation with the card security presence value set to the PIN_FLD_AVAILABLE value.
- If only the PIN_FLD_AVAILABLE value is present in the input flist, Paymentech DM sends the FR or PFR record for validation with the card security presence value set to the PIN_FLD_AVAILABLE value.
- If both PIN_FLD_SECURITY_ID and PIN_FLD_AVAILABLE values are not present in the input flist, FR or PFR record is not sent for validation.

### Card Security Code Is Now Masked in Logs

BRM applications may log flists containing the sensitive customer data. In previous releases, the card security code, such as VISA CVV2 or American Express CID, passed in the PIN_FLD_SECURITY_ID input flist field was not masked and appeared as clear text in the logs.

With this enhancement, the PIN_FLD_SECURITY_ID value is masked during logging. The card security codes passed in this field appear as masked fields in logs.

### Display Bills Generated Before Moving the Account to a Hierarchy

In previous releases, when an account was moved to a hierarchy as a child account, the bills generated for the account before moving it to the hierarchy were not displayed in the bills list in Billing Care.

BRM opcodes have been modified to retrieve and display all the bills for an account in hierarchy. This allows you to view the bills that are generated before and after moving the account to a hierarchy with the details (such as item list, event details, A/R actions, and payments) in Billing Care.

To support this feature, the following opcodes now include the new PIN_INCLUDE_CHILDREN_ALL value for the PIN_FLD_INCLUDE_CHILDREN parameter in the input flist:

- PCM_OP_AR_GET_BILLS
When you set the value of PIN_FLD_INCLUDE_CHILDREN to 3 (PIN_INCLUDE_CHILDREN_ALL), all the bills generated for the child account (before and after moving to the hierarchy) are retrieved.

Note:

When the PIN_FLD_INCLUDE_CHILDREN parameter is set to 3, PIN_FLD_BILLINFO_OBJ is mandatory in the input flist.

Managing Discount Validity Starting or Ending in Future Cycles

In previous releases, the discount proration was not set properly if the discount validity was starting or ending in a future cycle.

With this enhancement, during billing, BRM identifies the discounts that starts or ends in the next billing cycle and sets the discount validity and proration appropriately. For example, if proration for a discount is set to Full discount, full discount is applied even if the discount validity ends in the middle of the next billing cycle.

For more information on discount validity, see the discussion about configuring discount validity in PDC Creating Product Offerings if you are using PDC or the Pricing Center Online Help if you are using Pricing Center.

Wildcard in Item Type Selectors

Oracle Communications Pricing Design Center (PDC) now supports wildcard (*) in item type selectors. By setting the true or false value for the applicableToAllChildServices and applicableToAllChildEvents elements in PDC, you can configure whether all the child services or events of a service or event must be considered for item assignment.

If set to true, the real-time rating engine and batch rating engine consider all child services or events for the specified item assignment. If set to false, the real-time rating engine and batch rating engine do not consider child services or events for the specified item assignment.

If you are using wildcard in item type selectors, you must set the PDCEnable entry to true in the DAT_ItemAssign module in the Pipeline_home/conf/wireless.reg file.

For more information on using wildcard in item type selectors, see "PDC Now Supports Wildcard in Item Type Selectors".
BRM Supports POID Generation in ECE

In the previous releases, ECE was using the Portal object IDs (POIDs) generated in BRM for tracking rated events and bill items created in ECE.

With this enhancement, POIDs can be generated in ECE. ECE uses Rated Event Formatter to generate the required POIDs. To support this feature, the following changes have been made in BRM:

• All the existing BRM storable class definitions have been modified to include the event type. The event type is used for creating separate partitions for different set of events. When you create new custom classes in BRM, you must now set the **Event Type** field. For more information, see “About Creating Custom Classes”.

• The PCM_OP_SDK_SET_DD, PCM_OP_SET_DD, and PCM_OP_GET_DD opcodes have been modified to support the partitioning of prepaid events.

• The following have been introduced to enable partitions for prepaid events:
  – The **prepaid_partition_set** and **prepaid_partition_transition_mode** entries introduced in **dm_oracle**.
  – The **prepaidPartitionSet** parameter introduced in the **system** instance of the **/config/business_params** object.

  For more information, see “Enabling Prepaid-Event Partitions”.

• The **partition_utils** utility now supports the **-t prepaid** parameter. You can run the **partition_utils** utility with the **enable** operation and this parameter to create partitions for the prepaid events.

Enabling Prepaid-Event Partitions

If you are using ECE for usage rating, you must enable prepaid-event partition in BRM for generating the POIDs in ECE.

To enable prepaid-event partitioning:

1. Open the **BRM_Homelsys/dm_oracle/pin.conf** file in a text editor.

2. Set the **prepaid_partition_set** entry to a numerical value between 2 and 7. For example:

   - **dm prepaid_partition_set 2**

   If this entry is set to 0, ECE uses the POIDs received from BRM for the prepaid events.

3. Set the **prepaid_partition_transition_mode** entry to 1:
Note:

Setting this entry to 1 enables Data Manager to retrieve the partitions for the existing events. After retrieving all the partitions for the existing events (for example, after 90 days), set this entry to 0 to disable this mode.

- dm prepaid_partition_transition_mode 1

4. Save and close the file.

5. Create an editable XML file from the system instance of the /config/business_params object:

   `pin_bus_params -r BusParamsSystem bus_params_system.xml`

6. Set the prepaidPartitionSet parameter to the value you specified in step 2. For example:

   `<prepaidPartitionSet>2</prepaidPartitionSet>`

7. Save the file as `bus_params_system.xml`.

8. Load the XML file into the BRM database:

   `pin_bus_params bus_params_system.xml`

9. Stop and restart CM.

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

11. Go to the `BRM_home/apps/partition_utils` directory.

12. Create partitions for the prepaid events by running the following command:

    `partition_utils -o enable -t prepaid`

    For more information, see the discussion about partitioning database tables in BRM System Administrator's Guide.

    For information on enabling POID generation and prepaid-event partitions for ECE, see the discussion about POID generation in BRM Elastic Charging Engine Release Notes.

About Creating Custom Classes

When you create new custom classes in BRM, you must now set the Event Type field. Table 2-1 lists the event types in BRM. For instructions on how to create custom classes, see the discussion about creating custom classes in BRM Developer's Guide.
When you create a custom class, note the following:

- If the event type is set to NONE, the corresponding event is not synchronized with PDC.
- When you add a subclass, ensure that the event type matches the event type of the parent class if the event type is set to anything other than NONE.

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USAGE_PREPAID</td>
<td>Specifies the prepaid events (real-time charging events) from ECE.</td>
</tr>
<tr>
<td>USAGE_POSTPAID</td>
<td>Specifies the delayed events in BRM and postpaid events (offline charging events) from ECE.</td>
</tr>
<tr>
<td>SUBSCRIPTION_RECURRING</td>
<td>Specifies the recurring subscription event.</td>
</tr>
<tr>
<td>SUBSCRIPTION_ROLLOVER</td>
<td>Specifies the subscription events generated for rollovers.</td>
</tr>
<tr>
<td>SUBSCRIPTION_BILL_TIME_DISCOUNT</td>
<td>(For internal use only) Specifies the subscription events generated for bill-time discounts.</td>
</tr>
<tr>
<td>SUBSCRIPTION_FOLD</td>
<td>(For internal use only) Specifies the subscription events generated for folds.</td>
</tr>
<tr>
<td>SUBSCRIPTION_ONE_TIME</td>
<td>(For internal use only) Specifies the one-time subscription events.</td>
</tr>
<tr>
<td>SUBSCRIPTION_REMITTANCE</td>
<td>(For internal use only) Specifies the subscription events generated for remittances.</td>
</tr>
<tr>
<td>NONE</td>
<td>Specifies that the event does not belong to any other event type.</td>
</tr>
</tbody>
</table>

After you create the custom class and synchronize it with PDC, it is not recommended to change the event type of the custom class. However, if you have set the incorrect event type, you can change the event type by updating it first in BRM and then in PDC.

For changing the event type in BRM by editing the custom class, see the discussion about creating, editing, and deleting fields and storable classes in BRM Developer's Guide.

For updating the event type in PDC and publishing it to ECE, see the discussion about synchronizing and publishing the event type in "PDC Synchronizes Event Data Using Event Types".

Support for Stored-Credential Transactions for Payments

Credit card networks, such as VISA, MasterCard, Diners, Discover, JCB, and American Express, support the stored credential framework. They allow the merchants to use stored credentials for transactions. A stored credential is a payment information (such as an account number or a payment token) of a card holder stored by a merchant or its agent, a payment facilitator, or a staged digital wallet operator to
process future transactions for the card holder. The Paymentech card processors also support customer-initiated or merchant-initiated transactions with stored credentials.

With this enhancement, BRM supports payment transactions with stored credentials for VISA, MasterCard, Diners, Discover, JCB, and American Express cards. You can also customize BRM to support stored-credential transactions for other card networks. When VISA, MasterCard, Diners, Discover, JCB, and American Express cards are used for payments, the PCM_OP_PYMT_COLLECT opcode sends the following information required for card transactions to Paymentech Data Manager (dm_fusa) and stores the responses received from Paymentech DM for future transactions:

- Type of charge, such as recurring, one-time, and installment.
- Transaction type.
- Information on whether the card details are stored for future use.
- A unique ID (TXID) obtained from a previous verify/charge transaction of the same type.

You can override this information sent to Paymentech DM based on your business requirements. If you do not want to store credentials for future transactions, you can remove this information from the input passed to the PCM_OP_PYMT_COLLECT opcode.

For more information on storing or purging card credentials, see the following:

- Storing Card Credentials for Future Transactions
- Purging Card Credentials

If you already have payment information stored in BRM for cards that support the stored credential framework, see "Migrating Legacy Payment Information".

To support the stored credential framework, the following changes have been made in BRM for this feature:

- A new array, PIN_FLD_TRANSACTIONS, has been introduced in the /payinfo/cc storable class with the following optional fields to hold the Stored Credential Framework-specific information:
  - PIN_FLD_BILLINFO_OBJ. The bill unit (billinfo) for which payment is applied using the payment information (payinfo) in this array.
  - PIN_FLD_MODE. The message type with which the initial transaction was performed.
  - PIN_FLD_TRANS_ID. The TXID received in the initial transaction response.

- A new array, PIN_FLD_TRANSACTIONS, has been introduced in the /event/billing/charge/cc and /event/billing/validate/cc storable class with the following optional fields to hold the Stored Credential Framework-specific information:
  - PIN_FLD_TRANS_ID. The TXID received in the initial transaction response.
– **PIN_FLD_MODE.** The message type with which the transaction was performed.

– **PIN_FLD_FLAGS.** The flag which indicates whether the credentials are stored in the file.

– **PIN_FLD_TYPE.** The transaction type.

• The following opcodes have been modified to include the PIN_FLD_TRANSACTIONS array:
  – PCM_OP_PYMT_COLLECT
  – PCM_OP_PYMT_POL_PRE_COLLECT
  – PCM_OP_PYMT_CHARGE
  – PCM_OP_PYMT_CHARGE_CC
  – PCM_OP_CUST_COMMIT_CUSTOMER
  – PCM_OP_CUST_CREATE_PAYINFO
  – PCM_OP_CUST_SET_PAYINFO
  – PCM_OP_CUST_POL_VALID_PAYINFO

• The following tables have been added to store the information in the PIN_FLD_TRANSACTIONS array:
  – EVT_BILLING_CHARGE_CC_TRANS_T
  – EVT_BILL_VLDT_CC_TRANS_T
  – PAYINFO_CC_TRANS_T

• BRM payment collection utilities, such as `pin_collect` and `pin_deposit`, have been enhanced to support stored credentials.

For more information on the field definitions and valid values, see *BRM Opcode Flist Reference*.

### Storing Card Credentials for Future Transactions

To store the credit card information for future transactions, use PCM_OP_PYMT_COLLECT.

This opcode does the following:

1. Receives the PIN_FLD_TRANSACTIONS array from the following opcodes if a VISA, MasterCard, Diners, Discover, JCB, or American Express card is registered for payment:
   
- **PCM_OP_CUST_COMMIT_CUSTOMER.** This opcode passes the PIN_FLD_TRANSACTIONS array as input when you register a new customer with *Credit Card* as the default payment method and the cc_validate or cc_collect flag in the CM configuration file or credit card tokenization is enabled.

- **PCM_OP_CUST_SET_PAYINFO.** This opcode passes the PIN_FLD_TRANSACTIONS array as input when you set *Credit Card* as the default payment method for the account.
• **PCM_OP_CUST_CREATE_PAYINFO.** This opcode passes the PIN_FLD_TRANSACTIONS array as input when you add a new credit card and set that as the default payment method for the account.

2. Accepts the card information in the PIN_FLD_TRANSACTIONS array, adds missing information as required, and then passes the information as input to the PCM_OP_PYMT_POL_PRE_COLLECT policy opcode.

   **Note:**
   You can add, update, or remove the card information in the PIN_FLD_TRANSACTIONS array by customizing the PCM_OP_PYMT_POL_PRE_COLLECT policy opcode.

3. Receives the updated card information and PIN_FLD_CHARGES from the PCM_OP_PYMT_POL_PRE_COLLECT policy opcode.

4. Sends the card information in the PIN_FLD_TRANSACTIONS array to Paymentech DM by calling the PCM_OP_PYMT_CHARGE or PCM_OP_PYMT_CHARGE_CC opcode.

   Paymentech DM appends the required records based on the information received and sends the transactions to Paymentech. If the transaction is successful, Paymentech DM retrieves the TXID from the Paymentech response and passes it to the PCM_OP_PYMT_COLLECT opcode.

5. Accepts the TXID received from Paymentech DM and stores it in the PIN_FLD_TRANSACTIONS array in the `/payinfo/cc` object for future transactions.

### Purging Card Credentials

When **Credit Card** is set as the default payment method, the PIN_FLD_TRANSACTIONS array with the card information for each bill unit is stored in the `/payinfo/cc` object. If the default payment method is changed to any other method or card, the PIN_FLD_TRANSACTIONS array in the `/payinfo/cc` object are automatically purged from the database if the payments for the accounts are not in cardholder-initiated installments. If the payments are in cardholder-initiated installments, the card holder must delete the PIN_FLD_TRANSACTIONS array manually after the payment is made for the last installment.

### Migrating Legacy Payment Information

If there is legacy payment information in BRM for cards that support stored credential framework, it is automatically migrated to the new format, which includes the fields for storing credentials. BRM uses this information only for merchant-initiated transactions.

After migration, when the merchant initiates the first transaction with this payment information, the PCM_OP_PYMT_COLLECT opcode sets the TXID in the input as `EXISTING99999999` to indicate that this transaction is initiated with the legacy payment information. The opcode then stores the TXID received from Paymentech DM as an additional TXID.
Support for Migration of Legacy Data into BRM and ECE in Real Time

In previous releases, all the legacy data migrated to BRM were loaded into Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) asynchronously using CustomerLoader. In case if the ECE update failed, the BRM and ECE data would be unsynchronized. If ECE uses the unsynchronized data to rate usage events, the events might be rated incorrectly.

Also, it was not possible to migrate legacy service and balance data incrementally into BRM and ECE. For example, when the legacy data was migrated into BRM in phases, if the account already migrated to BRM had some services associated with it, you could not migrate additional legacy services for the same account. As a result, the legacy service and balance updates could not be loaded into ECE.

With this enhancement, you can migrate the legacy data completely or incrementally into the BRM system by using the pin_cnt utility and also load all the migrated data into ECE synchronously (in real time). For example, after the initial migration of the legacy account and service data into BRM and ECE, you can migrate additional services and balances for the same migrated accounts into BRM and also synchronize them with ECE in real time.

To support this feature, the following changes have been made in BRM:

- The CMT_Service.xsd schema file has been introduced to support incremental migration of legacy service data.
- The CMT_Balances.xsd schema file has been modified to support incremental migration of legacy balance data.
- A new entry, infranet.cmt.uselegacybalances, has been introduced in the BRM_Home/apps/cmt/Infranet.properties file to support incremental migration of legacy balance data.
- The cmt_mta_cycle_fees utility, which is run internally by the pin_cmt utility, has been modified and renamed as cmt_mta_deploy. Do not run this utility by itself.
- Following new parameters are introduced for migrating legacy data by using the pin_cmt utility:
  - deploy_ece specifies to deploy the migrated data in BRM and also load them into ECE synchronously. You can use this parameter for complete or incremental migration of legacy data into BRM and ECE synchronously.

  **Note:**

  If ECE is not integrated with BRM, you need to use the deploy parameter for both complete and incremental migration of legacy data into BRM. For more information, see the discussion about loading legacy data into the BRM database in BRM Migrating Accounts to the BRM Database.

  - deploy_db specifies to deploy the migrated data only in BRM. You can use this parameter if you want to load the legacy data into ECE asynchronously using CustomerLoader.
Note:

After deploying the legacy data by using the `deploy_db` parameter, you need to load the data manually into ECE by using the `CustomerLoader` utility and then run the `pin_cmt` utility with the `apply_cycle_fees` parameter to apply the cycle fees for the migrated accounts.

For information on loading data using the `CustomerLoader` utility, see the discussion about using `CustomerLoader` in *BRM Elastic Charging Engine Implementing Charging*.

- `apply_cycle_fees` specifies to apply the cycle fees for the migrated accounts. You can use this parameter only when you load the legacy data into ECE asynchronously.

When you use the `deploy_ece` parameter, the `pin_cmt` utility deploys the migrated legacy data in BRM and also sends it to ECE in real time through External Manager (EM) Gateway. In this case, both the BRM database and the ECE cache updates occur in a single transaction. If the ECE cache update succeeds, the updates are saved to the BRM database. If the cache update fails, the database updates are rolled back. This ensures that the BRM and ECE data remain synchronized whether the cache update succeeds or fails. You can use the `cnt.pinlog` files in BRM and error logs in ECE to troubleshoot the errors. For more information on troubleshooting Conversion Manager, see *BRM Migrating Accounts to the BRM Database*. For more information on troubleshooting ECE, see *BRM System Administrator’s Guide*.

To migrate the legacy account, service, and balance data completely or incrementally into BRM and ECE, do the following:

1. Import the legacy data into BRM. See "Importing Data".

2. Deploy the converted data. See "Deploying Data in BRM and ECE".

Importing Data

You import legacy data into the BRM database one file at a time.

Before importing the legacy data, do the following:

1. If you are migrating custom data, create new or extended storable classes for migration. See the discussion about migrating data by using new and extended storable classes in *BRM Migrating Accounts to the BRM Database*.

2. Create XML files with the legacy data that you want to import, which conform to the format detailed in the corresponding XSD files; such as `CMT_Account.xsd`, `CMT_Service.xsd`, and `CMT_Balances.xsd` file. See the discussion about creating XML files in *BRM Migrating Accounts to the BRM Database*.

3. Open the `BRM_home/apps/cmt/Infranet.properties` file in a text editor.

4. Ensure that the database connection is specified.

5. If you are migrating legacy balances, to replace the existing balances in BRM with the legacy balances, set the `infranet.cmt.uselegacybalances` entry to `true`. 
Note:

This entry takes precedence over the infranet.cmt.deleteexistingbalances entry. If set to true, the infranet.cmt.deleteexistingbalances entry is not used.

Note:

When the infranet.cmt.uselegacybalances entry is set to true, you must ensure the following:

- Check if the VALID_FROM date is specified for all the sub-balances to be migrated. The VALID_FROM date is required for identifying the sub-balances to be replaced.
- Import the legacy balances before deploying the account and service data to the BRM production area. You cannot replace the balances after you deploy the migrated account and service data.

6. Save and close the file.

To import the legacy data completely or incrementally into the BRM database:

1. Go to the BRM_home/apps/cmt directory.

2. Do one of the following:

   - If you have not created custom storable classes for migration, import the legacy data by running the following command:

     ```
     pin_cmt -import -file XML_input_data_file stage_ID
     ```

     where:

     - `XML_input_data_file` is the XML file with the legacy data.
     - `stage_ID` is the unique identity of the staging area.

   - If you have created custom storable classes for migrating custom data, import the legacy data by running the following command:

     ```
     pin_cmt -import_custom -file XML_input_data_file stage_ID
     ```

Note:

Ensure that the `stage_ID` is unique for each migration.

Deploying Data in BRM and ECE

Before deploying the imported data in BRM and ECE, do the following:

- Ensure that all the pricing data is loaded into the ECE cache. See the discussion about verifying that pricing data is loaded into ECE in BRM Elastic Charging Engine Implementing Charging.
• Ensure that the real-time synchronization is enabled in ECE. See the discussion about enabling real-time synchronization of BRM and ECE customer data updates in *BRM Elastic Charging Engine Implementing Charging*.

To deploy the imported data to the BRM production area and load the data into ECE synchronously, run the following command:

```
pin_cmt -deploy_ece DOM stage_ID
```

where *DOM* is the billing cycle's day of month.

The migrated legacy data deployed in BRM and are synchronized with ECE in real time.

If you are deploying the services, the *pin_cmt* utility does the following:

• Applies the cycle fees in BRM and synchronize them with ECE if the *infranet.cmt.deploy.opcode* entry is set to *true*. See the discussion about applying cycle fees to deployed accounts in *BRM Managing Customers*.

• Deploys the legacy balances if the *infranet.cmt.uselegacybalances* entry is set to *true*.

Merging Legacy Balances

BRM merges the migrated legacy balance data with the existing balance data in the BRM database. If the sub-balance already exists in the BRM database, it replaces the sub-balance with the corresponding legacy balance. If the sub-balance does not exist, it adds the legacy balance to the BRM database. BRM identifies the existing sub-balance in the BRM database by using the combination of the following data in *cmt_balances* objects:

• resource ID
• POID of the granting product/charge offer or discount
• VALID_FROM date

In case if the VALID_FROM date is not specified, the first sub-balance available for the specified granting product or discount is replaced with the legacy balance. However, you must ensure that the VALID_FROM date is specified for all the legacy balances to be migrated. This ensures that the balances are replaced appropriately.

In case if the POID of the granting product or discount is not specified, a new sub-balance is created with the specified VALID_FROM and VALID_TO dates.
For more information on migrating data, see the discussion about loading legacy data into the BRM database in *BRM Migrating Accounts to the BRM Database*.

**Support for Migrating Hierarchical Accounts Using Same Input File**

In previous releases, you could migrate a legacy hierarchical account group to BRM only if the parent and child accounts belonging to that group were present in different XML input files.

With this enhancement, you can migrate the legacy hierarchical account groups to BRM even if the parent and child accounts in the groups are present in the same XML input file. When you run the `pin_cmt` utility, all the hierarchical account groups in the input file are loaded into the BRM database. However, if any group in the file contains a multi-level account hierarchy (for example, Account C's bill unit is paid by Account B, which is in turn paid by Account A), you cannot load the parent and child accounts in that group using the same input file. You must create separate input files for parent and child accounts and load the parent accounts into the BRM database before loading the child accounts.

**Event Rounding Rules Can Be Used for Adjustments**

You can now use the rounding rules configured for events in adjustments. For example, if you configured usage events with six decimal precision, the event-level adjustment on the usage fee is rounded to six decimal precision.

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

To use the rounding rules of corresponding events for adjustments:

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

**New Staging Directory for Exported G/L reports**

In the previous releases, when the `pin_ledger_report` utility was run in the export mode, the utility created a staging directory named, `TEMP_XML_STAGING`, in the `OutputDirectory` and stored the output G/L report files temporarily in this directory before moving them to the `OutputDirectory`, where `OutputDirectory` is the directory on your system in which to create the output G/L report files.
With this enhancement, the `pin_ledger_report` utility creates a staging directory named, `TEMP_GL_REPORT_STAGING`, in the same location as the `OutputDirectory` and stores the output G/L report files temporarily in this directory before moving them to the `OutputDirectory`. For example, if the `OutputDirectory` is `pin/users/GL`, the G/L Report Files are stored in the new `pin/users/TEMP_GL_REPORT_STAGING` directory.

**Note:**
To enable the `pin_ledger_report` utility to store G/L report files in the new staging directory, do one of the following:

- Ensure that the utility has the permission to create a directory and file in the location in which the `OutputDirectory` is available.
- Manually create the `TEMP_GL_REPORT_STAGING` directory in the same location as the `OutputDirectory` and add the permission for the utility to create or store files in this directory.

### BRM Client Applications Supported from 12.0 Patch Set 2

The following table lists the BRM client applications that are now released in BRM 12.0:

**Table 2-2  Supported BRM Client Applications**

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer Center</td>
<td>BRM now supports Customer Center (including Customer Center SDK). You can now use Customer Center or Oracle Communications Billing Care for customer management tasks. For more information, see the Customer Center Online Help.</td>
</tr>
<tr>
<td>Collections Center</td>
<td>BRM now supports Collections Center. You can now use Collections Center or Billing Care for collections or debt management tasks. For more information, see the Collections Center Online Help.</td>
</tr>
<tr>
<td>Configuration Center</td>
<td>BRM now supports Configuration Center.</td>
</tr>
<tr>
<td>Field Validation Editor</td>
<td>BRM now supports Field Validation Editor to define how to validate customer data.</td>
</tr>
<tr>
<td>Payment Center</td>
<td>BRM now supports Payment Center. You can now use Payment Center or Billing Care instead for payment management tasks. For more information, see the Payment Center Online Help.</td>
</tr>
<tr>
<td>Payment Tool</td>
<td>BRM now supports Payment Tool. You can now use Payment Tool or Billing Care for processing payments. For more information, see the Payment Center Online Help.</td>
</tr>
<tr>
<td>Pricing Center</td>
<td>BRM now supports Pricing Center. You can now use Pricing Center or Pricing Design Center for creating product offerings for BRM. For more information, see the Pricing Center Online Help.</td>
</tr>
</tbody>
</table>

### New Features in BRM 12.0 Patch Set 1

BRM 12.0 Patch Set 1 includes the following enhancements:
• Improved Performance for Large Accounts
• Delay Interval Can be Configured for Resolving Failed Payments
• Enhanced Data Protection
• Enhanced Security for Root Wallet
• Support for Rolling Back the BRM Patch Set
• BRM 12.0 Is Now Certified with Mozilla Firefox 58.0
• BRM 12.0 Is Now Certified with Perl 5.28.0
• BRM 12.0 Is Now Certified with Paymentech 120 Byte Batch Version 3.0.0 R 12.4 and Online Authorization Version 7.4 R12.4
• BRM 12.0 Is Now Certified with Tomcat 8.5.32

Improved Performance for Large Accounts

Wholesale business accounts with large account hierarchies can have a large number of services each representing a subscription account. This can affect the billing and invoicing performance if the accounts had a large number of billing items to be processed.

With this enhancement, you can setup wholesale billing for handling large wholesale business accounts. In wholesale billing, you set up a bill unit hierarchy for account receivable (A/R) operations. In this hierarchy, the wholesale business account is the parent account with the paying parent bill unit and the services (subscriptions) in this account are child accounts with nonpaying child bill units. This enables BRM to consolidate the charges, discounts, A/R items, bill items, journals, and taxes across the services under the wholesale business account and perform the A/R operations, billing, and invoicing at the wholesale business account level. This improves the billing and invoicing performance for wholesale accounts with large hierarchies.

If you want to enable wholesale billing for all your accounts, you can enable system-wide wholesale billing by setting the WholesaleBillingSystem business parameter in the billing instance of the /config/business_params object. When this business parameter is enabled, you can create only wholesale accounts and bill unit hierarchies. For more information, see "Enabling Wholesale Billing" and "Creating Wholesale Accounts and Bill Unit Hierarchies".

If you want to enable wholesale billing only for specific accounts, you can set up an account with the paying parent bill unit as the wholesale parent and then create the wholesale bill unit hierarchy. You can create multiple wholesale bill unit hierarchies based on your business requirements. You need not enable system-wide wholesale billing. For more information, see "Creating Wholesale Accounts and Bill Unit Hierarchies".

Enabling Wholesale Billing

To enable wholesale billing system-wide:

1. Go to BRM_home/sys/data/config.
2. Create an XML file from the /config/business_params object:
   
   ```
   pin_bus_params -r BusParamsBilling bus_params_billing.xml
   ```
3. In the XML file, set the following entry to enabled:
4. Save the file as `bus_params_billing.xml`.

5. Load the file into the BRM database:
   ```
   pin_bus_params bus_params_billing.xml
   ```

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

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

Creating Wholesale Accounts and Bill Unit Hierarchies

You can create accounts and bill unit hierarchies for wholesale billing by using Billing Care or by using custom applications to call BRM opcodes. A wholesale bill unit hierarchy is headed by a paying parent bill unit with nonpaying child bill units beneath it. You can have only one paying bill unit in a wholesale bill unit hierarchy. And, you cannot add more than one bill unit to a wholesale parent account.

For more information on accounts and bill unit hierarchies, see the discussion about creating and managing account and bill unit hierarchies in `BRM Managing Accounts Receivable`.

>Note:
You cannot convert an existing bill unit hierarchy in your system into a wholesale bill unit hierarchy. You must ensure the following for enabling wholesale billing:

- If you are enabling system-wide wholesale billing, you must set the `WholesaleBillingSystem` business parameter before creating wholesale accounts and bill unit hierarchies.
- If you are enabling wholesale billing for specific accounts, you must set up a wholesale parent before creating the wholesale bill unit hierarchy.

To create accounts and bill unit hierarchies and set up the wholesale parent account by:

- Using Billing Care, see the discussion about creating accounts and configuring billing hierarchies in the Billing Care Online Help.
- Using BRM opcodes, see the discussion about creating accounts in the `BRM Opcode Guide` and "Setting Up a Wholesale Parent".

>Note:
Before you set up the wholesale parent either by using Billing Care or BRM opcodes, you must configure the wholesale business profile. See "Configuring Wholesale Business Profile".
You can add any existing bill unit to a wholesale bill unit hierarchy or set up a new wholesale bill unit hierarchy by using the existing bill units in BRM. However, you must ensure the following:

- There are no pending items or payments in the bill unit that you are adding to the hierarchy.
- The parent bill unit is the paying bill unit and it is set as the wholesale parent for billing.
- The wholesale parent for the wholesale bill unit hierarchy is set before creating the hierarchy.

This ensures that the charges and other billing-related items of the nonpaying child bill units in the hierarchy are rolled up to the paying parent bill unit during billing.

Configuring Wholesale Business Profile

To configure the wholesale business profile:

1. Open the `pin_business_profile.xml` file in an XML editor or a text editor. By default, this file is in the `BRM_home/sys/data/config` directory.
2. Search for the corporate wholesale business profile.
3. Set the following entry to **yes**:
   
   ```xml
   <WholesaleBilling>yes</WholesaleBilling>
   ```
4. Save and close the file.
5. Load the `pin_business_profile.xml` file by running the following command:
   ```
   load_pin_business_profile pin_business_profile.xml
   ```

   **Note:**

   - When you run the utility, the `pin_business_profile.xml` and `business_configuration.xsd` files must be in the same directory. By default, both files are in `BRM_home/sys/data/config`.
   - This utility needs a configuration (`pin.conf`) file in the directory from which you run the utility.
   - If you do not run the utility from the directory in which `pin_business_profile.xml` is located, include the complete path to the file. For example:
     ```
     load_pin_business_profile BRM_home/sys/data/config/pin_business_profile.xml
     ```
6. (Multischema systems only) Run the `pin_multidb` script with the `-R CONFIG` parameter. For more information, see *BRM System Administrator’s Guide*.
Setting Up a Wholesale Parent

To set up a parent bill unit for wholesale billing, you assign the bill unit that you want use as the wholesale parent to the wholesale business profile. To configure the wholesale business profile, see "Configuring Wholesale Business Profile".

You can assign the bill unit (/config/business_profile object) during or after account creation:

- To assign the bill unit to a business profile during account creation, see the discussion about assigning bill units to business profiles in BRM Opcode Guide.
- To assign the bill unit of an existing account to a business profile, see the discussion about changing a bill unit's business profile in BRM Opcode Guide.

Rolling Charges Up to the Wholesale Parent

**Note:**

In BRM 12.0 and BRM 12.0 Patch Set 1, the usage charges calculated by ECE and Pipeline Manager are not considered for wholesale billing; for example, the usage charges for telephony services.

During final billing, all the charges (such as recurring, purchase, and usage charges) applied to the nonpaying child bill units (wholesale child accounts) are aggregated based on the item-tag-to-item-type mapping (item configuration) and are rolled up to the corresponding bill items of the paying parent bill unit (wholesale parent account).

If the bill item for any item type does not exist for the paying parent bill unit, the bill item is created during billing and the charges are rolled up to that item. However, Oracle recommends to pre-create the bill items for the different item types by setting the precreate element to true in the BRM_home/sys/data/pricing/example/config_item_types.xml file. For more information, see the discussion about mapping item tags to item types in BRM Configuring and Running Billing.

The total and due amounts of the paying parent bill unit are updated to reflect the roll-up and the due amount of each nonpaying child bill unit is set to 0. Payments are applied only to the paying parent bill unit. From BRM 12.0 Patch Set 2, invoicing of due amount of child bill unit is supported. See "Wholesale Billing Enhancements".

If you want the charges for different services to be rolled up to different bill items, you can assign different item types for different services. For example, for rolling up cycle forward fees for IP and GSM services, you can configure and assign the following items: /item/ip/cycle_forward for the IP service and /item/gsm/cycle_forward for the GSM service.

You can also assign a different item type (or a noncumulative bill item) to track charges specific to the paying parent bill unit.

For information on assigning items, see the discussion about using event and service combinations to assign bill items in BRM Configuring and Running Billing.
Rolling A/R Actions Up to the Wholesale Parent

To manage balances for the A/R actions, BRM uses A/R items. A/R items include adjustment, dispute, settlement, payment, refund, payment reversal, write-off, and write-off reversal items. For more information, see the discussion about A/R management in BRM Concepts.

To roll up A/R actions for wholesale billing, you use the pin_roll_up_ar_items utility. The pin_roll_up_ar_items utility processes all the temporary A/R items (/tmp_ar_item_to_roll_up object) of the nonpaying child bill units and rolls the balance impact up to the corresponding A/R items of the paying parent bill unit. For example, this utility rolls the adjustments allocated to the nonpaying child bill unit's item/cycle_forward item up to the adjustment item associated with the item/adjustment item of the paying parent bill unit.

You can run the pin_roll_up_ar_items utility on a daily basis to ensure that the A/R items of the paying parent bill unit are kept up to date. However, you must run this utility once before billing the paying parent bill unit. For more information on this utility, see "pin_roll_up_ar_items".

In addition, you can use the pin_roll_up_ar_items utility to roll up the adjustment items that are created as a result of rerating. During rerating, the temporary A/R items (/tmp_ar_item_to_roll_up object) are created for the nonpaying child bill units if the following conditions are met:

- The event has already been billed.
- The event occurred prior to general ledger posting.
- The event is unbilled but the automatic allocation of rerating adjustments is disabled.

If the event is unbilled and the automatic allocation of rerating adjustments is enabled, the rerating adjustment is allocated to the bill item of the nonpaying child bill unit.

Note:

Rerating adjustments rolled up to the paying parent bill unit are allocated to the corresponding A/R item only if it exists in the paying parent bill unit. If the A/R item does not exist, the rerating adjustments remain unallocated at the parent level.

Rolling Journals Up to the Wholesale Parent

For nonpaying child bill units, the /tmp_journals_to_process objects are created instead of the /journal objects at the time of rating. The /tmp_journals_to_process objects are created only if the cycle_tax_interval entry in the CM configuration file is set to billing. For more information, see the discussion about tax calculation for account groups in BRM Calculating Taxes.

The /tmp_journals_to_process objects contain revenue and tax data. For wholesale billing, BRM uses these objects primarily to track and consolidate taxes for billing-time taxation. To roll up journals for wholesale billing, you use the pin_update_journal utility.
To roll up journals for wholesale billing, you must ensure that the general ledger reporting is enabled. For more information, see the discussion about general ledger reporting in *BRM Collecting General Ledger Data*.

The **pin_update_journal** utility processes all the `/tmp_journals_to_process` objects of the nonpaying child bill units and rolls them up to the corresponding `/journal` object of the paying parent bill unit.

You can run the **pin_update_journal** utility on a daily basis to ensure that the paying parent bill unit is kept up to date. However, you must run this utility once before billing the paying parent bill unit. For more information on this utility, see "**pin_update_journal**".

If deferred taxation is configured to consolidate taxes into a single item (if `cycle_tax_interval` is set to `billing`), the **pin_update_journal** utility enables you to roll the taxes up into a single item for both the paying parent and nonpaying child bill units. The total tax is calculated at the paying parent level for the entire hierarchy using the aggregated total due as the basis.

If deferred taxation is configured to calculate taxes separately for the parent and each nonpaying child bill unit (if `cycle_tax_interval` is set to `accounting`), the `/journal` objects are created for the nonpaying child bill units instead of `/tmp_journals_to_process` objects and the taxes are not rolled up to the paying parent bill unit.

### Running Wholesale Billing

**Note:**

The following features are not supported for wholesale billing:

- Bill Now
- On-purchase (on-demand) billing
- Skipped billing

If you are using bill suppression for billing wholesale accounts, you must run wholesale billing at the end of each accounting cycle.

To run wholesale billing:

1. Run the **pin_roll_up_ar_items** utility which rolls A/R actions up to the paying parent bill unit. See "**pin_roll_up_ar_items**".
2. Run the **pin_update_journal** utility which rolls journals up to the paying parent bill unit. See "**pin_update_journal**".
3. Run the **pin_bill_accts** utility which performs regular billing. See the discussion about the **pin_bill_accts** utility in *BRM Configuring and Running Billing*.
4. Run the `pin_inv_accts` utility for the wholesale parent account to generate invoices for bills.

**Configuring Billing Delay for Wholesale Hierarchies**

Delayed billing is supported for wholesale hierarchies. You must specify the billing delay even if it is not used. In this case, you can set the billing delay interval to 0. See the discussion about configuring billing delay in *BRM Configuring and Running Billing*.

**Setting Up Billing-Time Discounts for Wholesale Hierarchies**

For wholesale bill unit hierarchies, you set up a billing-time discount as follows:

- Configure billing-time discount only for the paying parent bill unit in the hierarchy.
- Configure BRM to apply the billing-time discount at the end of the billing cycle instead of the accounting cycle.
- Configure non-billing-time discounts (usage discounts) for the nonpaying child bill units in the hierarchy. The usage discount increments the counter. For example, if the billing-time discount for the paying parent bill unit is based on total monthly charges, you can create a discount for a nonpaying child bill unit that increments the counter when charges are applied.

And, for rolling the discounts up to the paying parent bill unit at the time of billing, you customize the `PCM_OP_SUBSCRIPTION_POL_PRE_CYCLE_DISCOUNT` policy opcode to return the list of balance element/resource IDs of the counters (in the `PIN_FLD_BALANCES` output flist field) for which the balances to be rolled up to the paying parent bill unit in the hierarchy.

For more information on billing-time discounts, see the discussion about creating discount offers in *BRM Creating Product Offerings*.

** Suppressing Bills for Wholesale Hierarchies**

You can use bill suppression to postpone finalizing bills for wholesale accounts. When bill suppression is enabled, the charges applied to the nonpaying child bill units are rolled up to the paying parent bill unit at the end of the accounting cycle. Therefore, you must run the `pin_roll_up_ar_items`, `pin_update_journal`, and `pin_bill_accts` utilities in the same order at the end of each accounting cycle. See "Running Wholesale Billing".

**Trial Billing for Wholesale Hierarchies**

When you perform trial billing for wholesale bill unit hierarchies, you must run the billing for nonpaying child bill units (*wholesale child accounts*) before running the billing for the paying parent bill unit (*wholesale parent account*). You can perform this by running the `pin_trial_bill_accts` utility with the `-pay_type` parameter. For more
information, see the discussions about trial billing for bill unit hierarchies and sharing groups and running trial billing according to payment type in BRM Configuring and Running Billing.

Support for A/R Activities

For wholesale bill unit hierarchies, the support for A/R activities varies from level to level:

- **Account level.** Adjustments, disputes, and settlements can be performed only at the parent bill unit level after billing.
- **Bill level.** Adjustments, disputes, and settlements can be performed only after billing at the parent bill unit level.
- **Event level.** Adjustments, disputes, and settlements can be performed at the child bill unit level before and after billing.
- **Item level.** Adjustments and disputes can be performed at the child bill unit level only before billing. After billing, adjustments and disputes are allowed only at the parent bill unit level. However, settlements can be performed before and after billing at the child bill unit level.

Write-offs can be performed only at the parent bill unit level after billing.

Specifying Search Criteria for Retrieving Items, Events, and Bills

To retrieve a list of items, events, or bills, BRM uses the following A/R and payment opcodes:

- PCM_OP_AR_GET_ACTION_ITEMS
- PCM_OP_AR_GET_ACCT_ACTION_ITEMS
- PCM_OP_AR_GET_BAL_SUMMARY
- PCM_OP_AR_GET_ACCT_BAL_SUMMARY
- PCM_OP_AR_GET_BILL_ITEMS
- PCM_OP_AR_GET_BILLS
- PCM_OP_AR_GET_DISPUTEDETAILS
- PCM_OP_AR_GET_DISPUTES
- PCM_OP_AR_GET_ACCT_BILLS
- PCM_OP_PYMT_ITEM_SEARCH
- PCM_OP_PYMT_MBI_ITEM_SEARCH
- PCM_OP_PYMT_SELECT_ITEMS

Based on the search criteria provided as input, these opcodes search all the bill units in a hierarchy. This can have an impact on the wholesale billing performance if you have large wholesale bill unit hierarchies.

To improve the wholesale billing performance, you must restrict the search to find items only for the specific bill units instead of searching all the bill units in a wholesale bill unit hierarchy.
Similarly, BRM uses the PCM_OP_AR_GET_ITEM_DETAIL and PCM_OP_AR_GET_ITEMS opcodes to retrieve the details of an A/R item or bill item for a bill unit. For wholesale hierarchies, these opcodes cannot retrieve all the details about the rolled-up items. For example, for the A/R items of the paying parent bill unit, these opcodes cannot retrieve the corresponding transfer events for the rolled-up disputes and settlements. Therefore, you must modify the search to retrieve only the details that are available for wholesale hierarchies.

For information on the search criteria for these opcodes, see *BRM Opcode Guide*.

### Moving Bill Units into or out of Wholesale Hierarchies

You can move the nonpaying child bill units into or out of a wholesale bill unit hierarchy. You can also move them between wholesale bill unit hierarchies. Before moving a bill unit, ensure that there are no pending items or payments in that bill unit.

When you move the nonpaying child bill unit to another hierarchy, all the items in that bill unit are associated with the corresponding items of the new paying parent bill unit and the new parent is billed for them. If a bill item does not exist in the new parent, it is created and the charges are rolled up to that item.

### Configuration Changes

You set the **StagedBillingFeeProcessing** business parameter to 4 to enforce cycle fee processing prior to billing and apply cycle forward fees in parallel by service with service charges aggregated to a single account item. With this enhancement, this option is no longer required. Now, the valid values for this parameter are only 0, 1, 2, and 3.

### Opcode Changes

**Opcode Changes**

To support wholesale billing, the following opcode changes have been made:

- The following new public opcode has been introduced:
  - PCM_OP_SUBSCRIPTION_POL_PRE_CYCLE_DISCOUNT
- The following public opcodes have been modified:
  - PCM_OP_AR_ACCOUNT_ADJUSTMENT
  - PCM_OP_AR_ACCOUNT_WRITEOFF
  - PCM_OP_AR_BILL_ADJUSTMENT
  - PCM_OP_AR_BILL_DISPUTE
  - PCM_OP_AR_BILL_SETTLEMENT
  - PCM_OP_AR_BILL_WRITEOFF
  - PCM_OP_AR_EVENT_ADJUSTMENT
  - PCM_OP_AR_EVENT_DISPUTE
  - PCM_OP_AR_EVENT_SETTLEMENT
For more information, see "Opcode Changes".

Schema Changes

To support wholesale billing, the following schema changes have been made:

- The following tables are added:
  - EVENT_ACT_ROLLUP_ITEMS_T
  - TMP_AR_ITEM_TO_ROLL_UP_T
- The following columns are added in the ITEM_T table:
  - ITEM_CLASS
  - AR_ITEM_OBJ
- The following column is added in the TMP_JOURNALS_TO_PROCESS_T table:
  - AR_BILLINFO_OBJ
- The following indexes are added:
  - I_TMP_AR_ITM_ROLLUP__ID
  - I_TMP_AR_ITM_ROLLUP__STATUS
  - I_ITEM_AR_ITEM_OBJ__ID
  - I_TMP_JOURNALS_TO_PROCESS__AR

Storable Class Changes

To support wholesale billing, the following changes have been made:

- The following new storable classes have been introduced:
  - /event/activity/roll_up
The following storable class has been modified:

- `/tmp_journals_to_process`

For more information, see "Storable Class Changes".

Utility Changes

To support wholesale billing, the following new utilities have been introduced:

- `pin_roll_up_ar_items`
- `pin_update_journal`. This utility replaces the `pin_update_items_journals` utility.

**pin_update_journal**

Use the `pin_update_journal` utility to process the temporary journals (`/tmp_journals_to_process` object) of the nonpaying child bill units and roll them up to the paying parent bill unit. You must run this utility before billing the paying parent bill unit.

To connect to the BRM database, the `pin_update_journal` utility needs a configuration file in the directory from which you run the utility. See the discussion about connecting BRM utilities in *BRM System Administrator's Guide*.

**Location**

`BRM_home/bin`

**Syntax**

`pin_update_journal [-verbose] [-help]`

**Parameters**

- **verbose.** Displays information about successful or failed processing as the utility runs.
- **help.** Displays the syntax and parameters for this utility.

**Results**

If the `pin_update_journal` 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.

**Error Handling**

When the `pin_update_journal` utility encounters an error while processing the A/R items in the temporary tables, it sets the PIN_FLD_BILLING_STATUS billing status field of the paying parent bill unit (`/billinfo` object) to PIN_BILL_ERROR. In addition, it sets the appropriate bit of the PIN_FLD_BILLING_STATUS_FLAGS field of the `/billinfo` object as PIN_BILL_FLAGS_UPDATE_JOURNALS_ERROR (bit value 0x2000).

After you have resolved the processing errors, you can reprocess the A/R items by running the `pin_update_journal` utility again.
pin_roll_up_ar_items

Use the pin_roll_up_ar_items utility to process the temporary A/R items (`tmp_ar_item_to_roll_up` object) of the nonpaying child bill units and roll the balance impact up to the corresponding A/R items in the paying parent bill unit. You can run multiple threads of pin_roll_up_ar_items to process A/R items for different paying parent bill unit.

You must run this utility before billing the paying parent bill unit. For more information, see "Rolling A/R Actions Up to the Wholesale Parent".

To connect to the BRM database, the pin_roll_up_ar_items utility needs a configuration file in the directory from which you run the utility. See the discussion about connecting BRM utilities in BRM System Administrator's Guide.

Location

BRM_home/bin

Syntax

pin_roll_up_ar_items [-verbose][-help]

Parameters

• verbose. Displays information about successful or failed processing as the utility runs.
• help. Displays the syntax and parameters for this utility.

Results

If the pin_roll_up_ar_items 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.

Error Handling

When the pin_roll_up_ar_items utility encounters an error while processing the A/R items in the temporary tables, it sets the PIN_FLD_BILLING_STATUS billing status field of the paying parent bill unit (`billinfo` object) to PIN_BILL_ERROR. In addition, it sets the appropriate bit of the PIN_FLD_BILLING_STATUS_FLAGS field of the `billinfo` object as PIN_BILL_FLAGS_UPDATE_ITEMS_ERROR (bit value 0x4000).

After you have resolved the processing errors, you can reprocess the A/R items by running the pin_roll_up_ar_items utility again.

Delay Interval Can be Configured for Resolving Failed Payments

In previous releases, when the pin_recover utility and a custom application were run in parallel to resolve failed credit card or debit card payments, duplicate transaction IDs were created for such transactions.

With this patch, a new entry, event_search_delay, has been introduced in the BRM_home/apps/pin_billd/pin.conf file to specify the delay interval for resolving failed payments. You can now run the pin_recover utility with event_search_delay to
delay the search of events so that `pin_recover` processes only the events (event/billing/charge/cc) that were created before the specified delay interval.

To configure delay interval for resolving failed payments:

1. Open the billing utility configuration file (`BRM_homedapps/pin_billd/pin.conf`) in a text editor.
2. Search for the `event_search_delay` entry.
3. Specify the delay interval:
   
   ```
   pin_recover event_search_delay value
   ```

   where `value` is the delay interval in seconds. By default, it is set to 0.

   For example, setting the `event_search_delay` entry to 300 delays the event search for resolving failed payments by 5 minutes:

   ```
   pin_recover event_search_delay 300
   ```

4. Save and close the file.

**Enhanced Data Protection**

BRM now includes the following security enhancements to protect the subscriber's personal data:

- Deleting closed accounts and all the related objects (such as events, items, bills, invoices, journals, newsfeeds, and user activities) and the audit data automatically from BRM after the specified retention period.

- Purge deleted accounts and the associated customer data synchronously on BRM and Oracle Communications Elastic Charging Engine (ECE). For more information, see the discussion about purging accounts from the ECE cache in `BRM ECE Release Notes`.

- Securing communications between BRM applications and the database. See the discussion about configuring SSL for the BRM database in `BRM 12.0 Patch Set Installation Guide`.

To support the security enhancements, the following changes have been made in BRM:

- The `ClosedAcctsRetentionMonths` business parameter has been introduced to specify the retention period for the closed accounts. See "Specifying Retention Period for Closed Accounts".

- The `pin_del_closed_accts` utility has been introduced to delete closed accounts from BRM after the specified retention period. See "pin_del_closed_accts".

- The `PCM_OP_CUST_DELETE_ACCT` opcode has been modified to ensure that all the BRM objects and audit entries containing the subscriber's personal data are purged.
Note:

You can use the PCM_OP_CUST_DELETE_ACCT opcode to delete accounts in a production system, but ensure that you use this opcode with care.

For more information on the PCM_OP_CUST_DELETE_ACCT opcode, see the discussion about deleting accounts in BRM Opcode Guide.

Note:

The PCM_OP_CUST_DELETE_ACCT opcode does not delete all the custom objects. You can write a custom logic to clean up the custom objects in BRM when the /event/notification/account/pre_delete and /event/notification/account/delete events are generated by the PCM_OP_CUST_DELETE_ACCT opcode.

Specifying Retention Period for Closed Accounts

You can specify the number of months the closed accounts must be retained in BRM by setting the ClosedAcctsRetentionMonths parameter in the customer instance of the /config/business_params object.

To specify the retention period for closed accounts:

1. Go to BRM_home/sys/data/config.
2. Create an XML file from the /config/business_params object:
   
   ```bash
   pin_bus_params -r BusParamsCustomer bus_params_customer.xml
   ```
3. Set the ClosedAcctsRetentionMonths entry to the number of months that you want to retain the closed accounts:
   
   ```xml
   <ClosedAcctsRetentionMonths>number_of_months</ClosedAcctsRetentionMonths>
   ```
4. Save the file as bus_params_customer.xml.
5. Load the XML file into the BRM database:
   
   ```bash
   pin_bus_params bus_params_customer.xml
   ```
6. Stop and restart the CM.
7. (Multischema systems only) Run the pin_multidb script with the -R CONFIG parameter. For more information, see BRM System Administrator's Guide.

Deleting Closed Accounts

You can delete the closed accounts in BRM after the retention period by using the pin_del_closed_accts utility.

To delete closed accounts:

1. Go to the BRM_home/apps/pin_billd directory.
2. Do the following as appropriate:

**Note:**
To delete all the closed child accounts in a hierarchy and the sharing groups you need to run the following commands in the following order.

- To delete all the closed nonpaying child accounts at different levels in a hierarchy, run the following command:
  
  \texttt{pin\_del\_closed\_accts -subord -leaf}
  
  \texttt{pin\_del\_closed\_accts -subord}

- To delete the member accounts of the sharing groups, run the following command:
  
  \texttt{pin\_del\_closed\_accts -members\_sharing}

- To delete the paying child accounts at different levels in a hierarchy, run the following command:
  
  \texttt{pin\_del\_closed\_accts -members\_billing}

**Note:**
You need to run this command for each paying account in a hierarchy. For example, if there are two paying accounts in the hierarchy, you must run this command twice to delete both the paying accounts.

3. Run the following command, which deletes all the remaining closed accounts, including the top-level parent account in the hierarchy:

\texttt{pin\_del\_closed\_accts}

4. If you want to delete specific closed accounts by using a file, run the following command:

**Note:**
Run the \texttt{pin\_del\_closed\_accts -file} command only if you want to delete specific accounts, but ensure that you use this command with care.

\texttt{pin\_del\_closed\_accts -file file\_name}

For example:

\texttt{pin\_del\_closed\_accts -file closed\_accts\_list.txt}

The utility deletes the accounts specified in the input file. You must provide the account details in the flist format in this file. For example:
pin_del_closed_accts

Use the pin_del_closed_accts utility to delete closed accounts from BRM. This utility calls the PCM_OP_CUST_DELETE_ACCT opcode to delete the closed accounts that are older than the specified retention period. For specifying the retention period, see "Specifying Retention Period for Closed Accounts".

To connect to the BRM database, the pin_del_closed_accts utility needs a configuration file in the directory from which you run the utility. See the discussion about connecting BRM utilities in BRM System Administrator's Guide.

Location

BRM_home/bin

Syntax

pin_del_closed_accts  
  -subord [-leaf]  
  -members_sharing  
  -members_billing  
  -file file_name  
  [-verbose] [-help]

Parameters

- **subord [leaf]**. Deletes the closed nonpaying child accounts at the bottom of the hierarchy.
- **subord**. Deletes the remaining closed nonpaying child accounts which are parents of other child accounts at the different levels of the hierarchy. Running the pin_del_closed_accts utility with this parameter does not delete the top-level parent account in the hierarchy.

You need to run the pin_del_closed_accts utility without any parameters after deleting all the paying and nonpaying child accounts at different levels in the hierarchy to delete the top-level parent account.

- **members_sharing**. Deletes the member accounts of the sharing groups; for example, discount and charge sharing groups.
- **members_billing**. Deletes the closed paying accounts in the hierarchy that are used for billing purposes.
- **file file_name**. Deletes the accounts specified in the input file. The file_name is the name and location of the file that contains the list of accounts for deletion. The account details in this file must be in the flist format.
Note:

Running the `pin_del_closed_accts` utility with this parameter deletes all the accounts specified in the input file even if the accounts are not older than the retention period. When you use this parameter, ensure that the input file contains only the closed accounts that need to be deleted.

- **verbose**. Displays information about successful or failed processing as the utility runs.
- **help**. Displays the syntax and parameters for this utility.

Results

The `pin_del_closed_accts` utility notifies you when it successfully deletes the closed accounts and the associated customer data.

If the `pin_del_closed_accts` 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.

After you have resolved the errors, you can delete the closed accounts by running the `pin_del_closed_accts` utility again.

Enhanced Security for Root Wallet

When you run the `pin_crypt_app` utility with the `-genrootkey` parameter, BRM now prompts for the root wallet password. This ensures that the root wallet is secured.

For more information, see the discussion about modifying the root encryption key in `BRM Developer's Guide`.

Support for Rolling Back the BRM Patch Set

BRM now allows you to roll back a BRM patch set. For example, if you experience issues after installing BRM 12.0 Patch Set 1, you can roll back BRM to 12.0.

For more information, see the discussion about rolling back a patch set in `BRM 12.0 Patch Set 1 Installation Guide`.

BRM 12.0 Is Now Certified with Mozilla Firefox 58.0

Currently, BRM 12.0 is certified with Mozilla Firefox 54.0.1. With this patch, BRM 12.0 is certified with Mozilla Firefox 58.0.

For more information, see `BRM Compatibility Matrix`.

BRM 12.0 Is Now Certified with Perl 5.28.0

Currently, BRM 12.0 is certified with Perl 5.24.0. With this patch, BRM 12.0 is certified with Perl 5.28.0.

For more information, see `BRM Compatibility Matrix`. 
BRM 12.0 Is Now Certified with Paymentech 120 Byte Batch Version 3.0.0 R 12.4 and Online Authorization Version 7.4 R12.4

Currently, BRM 12.0 is certified with Paymentech 120 Byte Batch Version 3.0.0 R 4.2 and Paymentech Online Authorization Version 7.4 R5. With this patch, BRM 12.0 is certified with Paymentech 120 Byte Batch Version 3.0.0 R 12.4 and Paymentech Online Authorization Version 7.4 R12.4.

For more information, see BRM Compatibility Matrix.

BRM 12.0 Is Now Certified with Tomcat 8.5.32

Currently, BRM 12.0 is certified with Tomcat version 8.5.16. With this patch, BRM 12.0 is certified with Tomcat version 8.5.32.

For more information, see BRM Compatibility Matrix.
New Features in ECE

This chapter provides an overview of the new features in Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) 12.0 Patch Set 2.

New Features in ECE 12.0 Patch Set 2

ECE 12.0 Patch Set 2 includes the following enhancements:

• New Tools to Monitor ECE
• Duplicate Check Enhancement in ECE
• Charging Operation Type Can be Configured for Expired Active Session Cleanup
• ECE Now Generates POID for Events
• New Tool for Querying ECE Cache Data
• ECE Now Supports Wildcard in Item Type Selectors
• Support for Persisting BRS Configuration Data
• Rated Event Partition in the ECE Persistent Database

New Tools to Monitor ECE

You can now use the following new monitoring tools to collect the ECE data for monitoring the ECE cache size, rating performance, latency, and so on:

• ece_cluster_overview. Collects the ECE cluster summary. Use this tool to display the summary in a text or comma-separated value (CSV) format in Linux.
• ece_full_gc. Collects the complete Garbage Collection (GC) debug logs for any ECE grid member or a specific java application by using the process ID.
• ece_log. Collects the specified log-level data. You can also use this script to set log levels. See "Modifying Log Levels by Using Scripts" for more information.
• ece_metrics_collector. Runs the ECE Metrics Collector tool. This tool starts a light-weight HTTP Server to collect the ECE metrics data and present the data to third-party monitoring tools. Use this tool to view the ECE metrics using open-source tools, such as Grafana and Prometheus. See "Monitoring ECE Using Grafana and Prometheus" for more information.
• jvm_info. Collects the JVM data on a periodic basis. The JVM data includes memory usage, central processing unit (CPU) usage, open files, and so on.
• ece_brs_info. Connects to a specific Java process in which the batch request service (BRS) is running and collects the data such as the number of requests processed and latency over time.
• ece_queues.sh. Collects Oracle Communications Pricing Design Center (PDC) metrics from WebLogic Server, such as the size of the pricing data, the number of messages in the queue, and so on.
• **cohql_file_processor.** Starts the query tool (**query.sh**) and allows it to remain connected to the ECE cluster. Handles the queries on ECE Coherence caches in the non-interactive or interactive mode. Use this script to query ECE data instead of running the **query.sh** script frequently.

• **count_ref_files.** Counts the number of Rated Event Formatter output files and their volume. You can use this script to estimate the volume of data that can be processed by Rated Event Formatter and the storage space required to store the rated event data in the event of failure.

• **get_log_slices.** Collects data from the specified set of ECE log files for the given time period to perform error analysis. You can use this script to collect the detailed information regarding a specific problem.

• **parse_ece_chronicler.** Parses the ECE batch request service (BRS) chronicler metrics to aggregate the data into configurable time periods. This script provides a general view of the ECE rating performance.

• **parse_ecedc.** Parses the Oracle Communications Offline Mediation Controller Elastic Charging Engine (ECE) Distribution Cartridge (DC) log files to collect the data for monitoring the rating performance and throughput of ECE DC deployed in Offline Mediation Controller; for example, batch processing time, data read time, batch submission time, timeouts, and so on.

• **parse_ece_pricing_load.** Parses the INFO-level log data in the Pricing Updater log files. This script provides the information required to monitor the Pricing Updater performance, such as the volume of data loaded, number of items processed, and the time taken to process the pricing data.

• **parse_ece_start.** Parses the INFO-level data in the ECE server log files generated during initial data loading. This script enables you to identify the performance problems in the initial data loading and startup process.

You can use these tools along with Coherence reports and ECE Monitoring Agent to monitor your ECE system. By default, the ECE monitoring tools are installed in the **ECE_home/tools** directory. If you change the location of these tools, run the following command in Bash shell, which sets the ECE_TOOLS_DIR and PATH environment variables to point to the new location:

```
export ECE_TOOLS_DIR=path_to_tools
export PATH=path_to_tools
```

For more information on these tools, see the following:

• Running ECE Monitoring Tools
• Customizing JMX-Based Tool Reports
• Monitoring ECE Cluster
• Monitoring ECE Using Grafana and Prometheus
• Setting Log Levels by Using Scripts
• Using cohql_file_processor for Query

**Running ECE Monitoring Tools**

You can run all these ECE monitoring tools with the default settings by using the following wrapper utilities:
Note:

You do not have to run each monitoring tool individually.

- **ece_util_launcher.sh.** Runs the following Groovy/JMX-based tools by using the default JMX credentials: `ece_cluster_overview`, `ece_metrics_collector`, `ece_brs_info`, `jvm_info`, `ece_log`, and `ece_full_gc`.

- **perl_util_launcher.sh.** Runs the following perl-based tools:
  - `cohql_file_processor`,
  - `count_ref_files`,
  - `get_log_slices`,
  - `parse_ece_chronicler`,
  - `parse_ecedc`,
  - `parse_ece Pricing_load`,
  - `parse_ece_start`.

You can also create symbolic link between tools to simplify the startup process. You can create symbolic links by running the following command in the `ECE_home/tools` directory or the directory in which the tools are available:

```
./mklinks all
```

For example, this creates symbolic links from `ece_cluster_overview` and `start_ece_metrics_collector` to the `ece_util_launcher.sh` and `perl_util_launcher.sh` utilities. This enables `ece_cluster_overview` and `start_ece_metrics_collector` to run the `ece_util_launcher.sh` and `perl_util_launcher.sh` utilities internally without any intervention.

Later, if you want to remove the symbolic link, run the following command:

```
./mklinks -r all
```

For the syntax and parameters used with the ECE monitoring tools, run the help (`-h`) command. For example:

```
ece_cluster_overview.groovy -h
```

Following is an example for running `ece_cluster_overview` to collect the cluster summary:

```
./ece_cluster_overview -e /scratch/ri-user-1/opt/OracleCommunications/ECE/ECE/oceceserver/config/eceTopology.conf -U controlRole -P 'R&d' -c
```

Customizing JMX-Based Tool Reports

The JMX-based tools generate reports in the tabular format. The output for these tools are defined in the `ECE_home/tools/TableDefinition.properties` file. You can customize the reports generated by these tools by updating this file or by using the filters when you run the tool command. For example, you can define the ECE attributes to be included in the reports and also change the width and format of the table based on your requirements.

For more information, see the `ECE_home/tools/TableDefinition.properties` file.

Monitoring ECE Cluster

The `ece_cluster_overview` tool connects to an ECE cluster through a JMX-enabled node and queries data about the cluster members and the state of ECE caches. This tool can be used to retrieve data on periodic basis in the text or comma-separated value (CSV) format.
You can also use this tool to query batch request service (BRS) data for charging clients, such as Diameter Gateway and Oracle Communications Offline Mediation Controller, and retrieve general JVM information, such as heap usage and CPU usage.

You can set the parameters used for collecting ECE cluster details by running the ece_cluster_overview.groovy script. For information on the parameters, default values, and examples, run the help (-h) command.

Monitoring ECE Using Grafana and Prometheus

You use the ECE Metrics Collector (ece_metrics_collector) tool that runs a lightweight HTTP server to collect system metrics from ECE on demand and convert the data into a format that can be processed by third-party monitoring tools. This tool provides different metric groups to collect different ECE metrics and present them to the third-party monitoring tools at different intervals.

You can use ECE Metrics Collector with the following third-party monitoring tools for monitoring ECE:

- **Prometheus.** An open-source monitoring tool which:
  - Stores system metrics in a time-series database.
  - Provides flexible queries to retrieve data for analysis.
  - Supports real-time alerting.
- **Grafana.** An open-source visualization tool that presents graphical dashboards by retrieving data from Prometheus. This tool contains multiple presentation formats and also provides an SDK for customizing dashboards.

The ECE monitoring process that uses Grafana and Prometheus is as follows:

1. ECE Metrics Collector collects the ECE metrics data in its different groups and presents them in the text-based exposition format to Prometheus by using HTTP endpoints.

   To configure, start, or stop ECE Metrics Collector, see "Starting and Stopping ECE Metrics Collector".

2. The Prometheus monitoring server connects to each HTTP endpoint in ECE Metrics Collector at different intervals periodically and queries the metrics data. For example, Prometheus collects the ECE cache metrics every 30 seconds and the usage rating metrics every 10 seconds.
To configure Prometheus for querying ECE metrics data, see the Prometheus documentation.

3. The Prometheus monitoring server stores the metrics data in a time-series database and runs rules on the metrics data to aggregate the data and generate alerts or notifications based on the configuration.

4. Grafana retrieves the ECE metrics data from the Prometheus monitoring server and presents different graphical dashboards for different users. The dashboard contains one or more panels which displays the metrics for ECE.

To configure Grafana for displaying the ECE metrics data, see the Grafana documentation.

For information on the metrics collected by ECE Metrics Collector, see "Collecting ECE Metrics Data".

Starting and Stopping ECE Metrics Collector

Before starting ECE Metrics Collector, you can set the ECE Metrics Collector parameters used for collecting ECE metrics by running the `ece_metrics_collector.groovy` script. For information on the parameters, default values, and examples, run the help (`-h`) command.

To start or stop ECE Metrics Collector:

1. On the driver machine, change to the `ECE_home/oceceserver/bin` directory.
2. Start ECC:
   ```bash```
   ./ecc
   ```
3. Do one of the following:
   - To start ECE Metrics Collector, run the following command:
     ```bash```
     start_ece_metrics_collector
     ```
     The HTTP server is started and it listens on the specified port. The default port is 5050.
   - To stop ECE Metrics Collector, run the following command:
     ```bash```
     stop_ece_metrics_collector
     ```
     The HTTP server is stopped.

You can verify if ECE Metrics Collector is running by connecting to the HTTP server using the following URL:

```
http://your_server:5050/metrics
```

Collecting ECE Metrics Data

ECE Metrics Collector collects metrics in different groups to produce data for monitoring ECE. See the following for more information:

- `ECE_GRID_MEMBER_METRICS`
- `BRS_CLIENT_METRICS`
- `CACHE_METRICS`
- `JVM_METRICS`
ECE_GRID_MEMBER_METRICS

The ECE_GRID_MEMBER_METRICS group contains the ece_grid_members metric which provides information about the running status of nodes in the topology. ECE Metrics Collector compares the information retrieved from the Coherence grid with the information in the ECE topology file to retrieve the list of all nodes in the grid with their status.

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_grid_members</td>
<td>Gauge</td>
<td>Contains the state of all the members in the ECE grid. The metric shows 1 for nodes that are running and 0 for nodes that are down.</td>
</tr>
</tbody>
</table>

For example, if the diameterGateway1 and ratedEventFormatter1 nodes are running and the query and Customer Loader utilities are down, the ece_grid_members metric retrieved is as follows:

```
ece_grid_members{id="21", pid="2313", machine="abc.us.oracle.com", nodeName="diameterGateway1", role="diameterGateway"} 1
ece_grid_members{id="22", pid="3239", machine="abc.us.oracle.com", nodeName="ratedEventFormatter1", role="ratedEventFormatter"} 1
ece_grid_members{id="0", pid="0", machine="abc.us.oracle.com", nodeName="CohqlShell", role="query"} 0
ece_grid_members{id="0", pid="0", machine="abc.us.oracle.com", nodeName="customerLoader", role="customerLoader"} 0
```

BRS_CLIENT_METRICS

The BRS_CLIENT_METRICS group contains the metrics for tracking throughput and latency of the charging clients, which use batch request service (BRS), such as Diameter Gateway, Offline Mediation Controller, or any custom charging clients.

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_brs_total_task_count</td>
<td>counter</td>
<td>Tracks the total number of requests processed by a charging client. This metric shows the number of requests accepted, processed, timed-out, or rejected by the ECE component. You can use this in Prometheus to track the approximate processing rate over time, aggregate over all client applications, and so on.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_brs_current_task_count</td>
<td>gauge</td>
<td>Tracks the number of requests processed by a charging client in the current scrape interval. This metric shows the number of requests accepted, processed, timed-out, or rejected by the ECE component since the last query. You can use this metric to determine the delta.</td>
</tr>
<tr>
<td>Metric Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>ece_brs_current_throughput_per_second</td>
<td>gauge</td>
<td>Tracks the throughput of a charging client for each charging operation type in the current query interval. For example, if the data is collected every 10 seconds, the current value is considered as the throughput for that specific operation type since the last query. This metric provides the throughput information for the following operation types: <em>Initiate</em>, <em>Update</em>, <em>Terminate</em>, <em>Cancel</em>, <em>Price_Enquiry</em>, <em>Balance_Query</em>, <em>Debit_Amount</em>, <em>Debit_Unit</em>, <em>Refund_Amount</em>, and <em>Refund_Unit</em>. You can use this metric to gauge the overall system throughput for each type of usage requests.</td>
</tr>
<tr>
<td>ece_brs_current_latency_milliseconds</td>
<td>gauge</td>
<td>Tracks the current operation latency (in milliseconds) for a charging client in the current scrape interval. This metric contains the BRS statistics tracked using the <code>charging.brsConfigurations</code> mbean attributes. This configuration tracks minimum, maximum, and average latency for an operation type since the last query. The maximum window size for collecting this data is 30 seconds, so the query has to be run within every 30 seconds. This metric provides the latency information for the following operation types: <em>Initiate</em>, <em>Update</em>, <em>Terminate</em>, <em>Cancel</em>, <em>Price_Enquiry</em>, <em>Balance_Query</em>, <em>Debit_Amount</em>, <em>Debit_Unit</em>, <em>Refund_Amount</em>, <em>Refund_Unit</em>, and <em>Spending_Limit_Report</em>.</td>
</tr>
<tr>
<td>ece_total_requests_by_result_code</td>
<td>counter</td>
<td>Tracks the total requests processed by Diameter Gateway by using the result code. This metric shows the total number of Gy and Sy requests processed by a Diameter Gateway instance for each result code.</td>
</tr>
<tr>
<td>ece_current_request_rate_milliseconds_by_result_code</td>
<td>gauge</td>
<td>Tracks the current processing rate of Diameter Gateway (in milliseconds) by using the Diameter protocols (Sy/Gy) and the result code. This metric provides the current processing rate for Gy and Sy operations handled by a Diameter Gateway instance for each result code.</td>
</tr>
</tbody>
</table>
Metric Name | Type | Description
--- | --- | ---
ece_session_metrics | gauge | Tracks the overall number of Gy sessions that are open or close for an ECE component, such as Diameter Gateway and ECE server nodes. This metric shows the total number of open and closed sessions for each Diameter Gateway instance and rating group. **Note:** By default, ECE Metrics Collector aggregates ECE server metrics by rating group at the system-level. You can refine this metric to include the information per ECS node by running ECE Metrics Collector with the `-mp` parameter.

**Example**

The following are the examples for the metrics in the BRS_CLIENT_METRICS group:

- **ece_brs_total_task_count.** This example shows that diameterGateway1 has processed 340845 requests without error and rejected or timed-out 0 requests:
  
  ```
ece_brs_total_task_count{nodeName="diameterGateway1", role="diameterGateway", machine="abc.us.oracle.com", type="accepted"} 340845
ece_brs_total_task_count{nodeName="diameterGateway1", role="diameterGateway", machine="abc.us.oracle.com", type="processed"} 340845
ece_brs_total_task_count{nodeName="diameterGateway1", role="diameterGateway", machine="abc.us.oracle.com", type="rejected"} 0
```

- **ece_brs_current_task_count.** This example shows that diameterGateway1 has accepted and processed 54 requests in the current scrape interval:
  
  ```
ece_brs_current_task_count{nodeName="diameterGateway1", role="diameterGateway", machine="abc.us.oracle.com" type="pending"} 0
```

- **ece_brs_current_throughput_per_second.** This example shows that diameterGateway1 has processed Initiate, Update, and Terminate (IUT) for voice calls at a rate of 30 calls per second:
  
  ```
ece_brs_current_throughput_per_second{nodeName="diameterGateway1", role="diameterGateway", machine="abc.us.oracle.com" opType="initiate"} 26
```

- **ece_brs_current_latency_milliseconds.** This example shows the minimum, maximum, and average latency in milliseconds for Initiate and Update requests:
  
  ```
```
• **ece_total_requests_by_result_code.** This example shows the total number of Gy and Sy requests processed by diameterGateway1 for the result code 2001:

```
ece_total_requests_by_result_code{nodeName="diameterGateway1", role="diameterGateway", machine="abc.us.oracle.com", protocol="Gy", resultCode="2001", resultCode="2001"} 237114
ece_total_requests_by_result_code{nodeName="diameterGateway1", role="diameterGateway", machine="abc.us.oracle.com", protocol="Sy", resultCode="2001", resultCode="2001"} 45883
```

• **ece_current_request_rate_milliseconds_by_result_code.** This example shows the current processing rate in milliseconds for Gy and Sy requests processed by diameterGateway1 for the result code 2001:

```
ece_current_request_rate_milliseconds_by_result_code{instance="abc.us.oracle.com:5050", job="ece_brs", machine="abc.us.oracle.com", nodeName="diameterGateway1", protocol="Gy", resultCode="2001", resultCode="2001"} 140.6
ece_current_request_rate_milliseconds_by_result_code{instance="abc.us.oracle.com:5050", job="ece_brs", machine="abc.us.oracle.com", nodeName="diameterGateway1", protocol="Sy", resultCode="2001", resultCode="2001"} 10.4
```

• **ece_session_metrics.** This example shows the total number of open and closed sessions for diameterGateway1 and the rating group 10:

```
ece_session_metrics{nodeName="diameterGateway1", rating_group="*", role="diameterGateway", type="Close"} 8140.0
ece_session_metrics{nodeName="diameterGateway1", rating_group="*", role="diameterGateway", type="Open"} 11750.0
ece_session_metrics{rating_group="10", type="Close"} 7639.0
ece_session_metrics{rating_group="10", type="Open"} 10645.0
```
CACHE_METRICS

The CACHE_METRICS group contains metrics for the ECE caches. You can use this metric to track the overall growth rate of certain caches along with other metrics.

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_cache_avg_size</td>
<td>Gauge</td>
<td>Contains the average size of objects (in bytes) in the ECE caches.</td>
</tr>
<tr>
<td>ece_cache_entries</td>
<td>Gauge</td>
<td>Contains the total number of entries present in an ECE cache on an ECE node at the time the query is run.</td>
</tr>
<tr>
<td>ece_cache_gets</td>
<td>Counter</td>
<td>Contains the total number of gets performed on an ECE cache.</td>
</tr>
<tr>
<td>ece_cache_puts</td>
<td>Counter</td>
<td>Contains the total number of updates made to an ECE cache. It is the mirror of the ece_cache_gets metric.</td>
</tr>
<tr>
<td>ece_cache_store_millis</td>
<td>Counter</td>
<td>Contains the cumulative time spent (in milliseconds) on the cache-store operations for an ECE cache. This metric is applicable only for the following caches: RatedEvent and ServiceContext.</td>
</tr>
<tr>
<td>ece_cache_store_writes</td>
<td>Counter</td>
<td>Contains the number of updates attempted on a cache store, such as RatedEvent and ServiceContext.</td>
</tr>
<tr>
<td>ece_cache_total_size</td>
<td>Gauge</td>
<td>Contains the total size of an ECE cache (by default in mega bytes). You can also define the unit of the metric by using the -u parameter.</td>
</tr>
</tbody>
</table>

Examples

The following are the examples for the metrics in the CACHE_METRICS group:

- **ece_cache_avg_size**. This example shows the average size in bytes for the ActiveSession, Customer, and RatedEvent caches:
  
  ```
  ece_cache_avg_size{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="ActiveSession"} 1968
  ece_cache_avg_size{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="Customer"} 3344
  ece_cache_avg_size{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="RatedEvent"} 2557
  ```

- **ece_cache_entries**. This example shows the total number of entries in the ActiveSession, Customer, and RatedEvent caches:

  ```
  ece_cache_entries{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="ActiveSession"} 3068
  ece_cache_entries{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="Customer"} 1117
  ece_cache_entries{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="RatedEvent"} 836
  ```

- **ece_cache_gets**. This example shows the total number of gets for the ActiveSession and Customer caches:

  ```
  ece_cache_gets{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="ActiveSession", tier="back"} 13906794
  ```
• **ece_cache_store_millis.** This example shows that 244.931 milliseconds and 5.019 milliseconds are spent on the RatedEvent and ServiceContext cache updates respectively:

\[
\text{ece_cache_store_millis\{nodeName=\"ecs1\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"RatedEvent\", tier=\"back\"\}\ 244.931}
\]

\[
\text{ece_cache_store_millis\{nodeName=\"ecs1\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"ServiceContext\", tier=\"back\"\}\ 5.019}
\]

• **ece_cache_store_writes.** This example shows the number of updates attempted for the RatedEvent and ServiceContext caches on 2 ECE charging server nodes:

\[
\text{ece_cache_store_writes\{nodeName=\"ecs1\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"RatedEvent\", tier=\"back\"\}\ 3182}
\]

\[
\text{ece_cache_store_writes\{nodeName=\"ecs1\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"ServiceContext\", tier=\"back\"\}\ 1}
\]

\[
\text{ece_cache_store_writes\{nodeName=\"ecs2\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"RatedEvent\", tier=\"back\"\}\ 2281}
\]

\[
\text{ece_cache_store_writes\{nodeName=\"ecs2\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"ServiceContext\", tier=\"back\"\}\ 0}
\]

• **ece_cache_total_size.** This example shows the size of the ActiveSession, Balance, and Customer cache in mega bytes:

\[
\text{ece_cache_total_size\{nodeName=\"ecs1\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"ActiveSession\", unit=\"mb\"\}\ 1.73}
\]

\[
\text{ece_cache_total_size\{nodeName=\"ecs1\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"Balance\", unit=\"mb\"\}\ 0.52}
\]

\[
\text{ece_cache_total_size\{nodeName=\"ecs1\", service=\"BRMFederatedCache\", machine=\"abc.us.oracle.com\", cache=\"Customer\", unit=\"mb\"\}\ 1.17}
\]

---

**JVM_METRICS**

The JVM_METRICS group contains standard metrics about the Central processing unit (CPU) and memory utilization of JVMs, which are members of the ECE grid.

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_jvm_memory</td>
<td>Gauge</td>
<td>Contains the Java heap usage information (by default in mega bytes) for each ECE charging server (ECS) node. This metric contains a lot of memory-related attributes of JVMs, which you can use to get a detailed view of the memory usage for different ECE components. <strong>Note:</strong> The off-heap memory data is tracked only if the <code>-M</code> parameter is not used while running the ECE Metrics Collector.</td>
</tr>
<tr>
<td>ece_jvm_process_cpu_percent</td>
<td>Gauge</td>
<td>Contains the CPU usage information (in percentage) for each ECE component on the server. This data is collected from the corresponding MBean attributes by JVMs.</td>
</tr>
<tr>
<td>ece_jvm_process_file_descriptors</td>
<td>Gauge</td>
<td>Contains the total number of file-descriptors currently available for an ECE component and the descriptors that are in use for that ECE component.</td>
</tr>
<tr>
<td>Metric Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>----------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ece_jvm_server_cpu_percent</td>
<td>Gauge</td>
<td>Contains the CPU load information (in percentage) for each system in the cluster. These statistics are based on the average data collected from all the ECE grid members running on a server.</td>
</tr>
<tr>
<td>ece_jvm_server_load_average</td>
<td>Gauge</td>
<td>Contains the system load average (the number of items waiting in the CPU run-queue) information for each machine in the cluster. These statistics are based on the average data collected from all the ECE grid members running on a server.</td>
</tr>
<tr>
<td>ece_jvm_server_swap</td>
<td>Gauge</td>
<td>Contains system swap usage information (by default in mega bytes) for each system in the cluster. These statistics are based on the average data collected from all the ECE grid members running on a server.</td>
</tr>
</tbody>
</table>

**Example**

The following are the examples for the metrics in the JVM_METRICS group:

- **ece_jvm_memory**. This example shows the values in mega bytes for a single ECS node:

  ```
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="heap", class="committed") 1536.00
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="heap", class="init") 1536.00
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="heap", class="percent") 42.56
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="heap", class="used") 653.79
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="mapped", class="count") 0.0
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="mapped", class="memory_used") 0.00
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="mapped", class="total_capacity") 0.00
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="off_heap", class="committed") 173.06
  ece_jvm_memory(nodeName="ecs1", role="server", machine="abc.us.oracle.com", type="off_heap", class="init") 2.44
  ```

- **ece_jvm_process_cpu**. This example shows that the 3 ECS nodes are using 6.41%, 6.31%, and 6.04% of CPU respectively:

  ```
  ece_jvm_process_cpu_percent(nodeName="ecs1", role="server", machine="abc.us.oracle.com") 6.409681757059616
  ece_jvm_process_cpu_percent(nodeName="ecs2", role="server", machine="abc.us.oracle.com") 6.308724832214765
  ece_jvm_process_cpu_percent(nodeName="ecs3", role="server", machine="abc.us.oracle.com") 6.037567084078712
  ```

- **ece_jvm_process_file_descriptors**. This example shows that BRM Gateway is using 144 descriptors and there are maximum of 65536 descriptors available for BRM Gateway:
**ece_jvm_process_file_descriptors**

- **tenername**="brmGateway", role="brmGateway",
  machine="abc.us.oracle.com", type="max" 65536
- **tenername**="brmGateway", role="brmGateway",
  machine="abc.us.oracle.com", type="open" 144

- **ece_jvm_server_cpu_percent**. The following example shows that the abc server is using approximately 31% of CPU:
  ece_jvm_server_cpu_percent{machine="abc.us.oracle.com"} 30.95

- **ece_jvm_server_load_average**. This example shows the abc server's load average is 3.08 at the time of query (if it is an 8 CPU sever, this indicates that the server is not heavily loaded):
  ece_jvm_server_load_average{machine="abc.us.oracle.com"} 3.08

- **ece_jvm_server_swap**. This example shows that abc server has 4096 mega bytes of swap in a total of 4 giga bytes and has about 2.8 giga bytes free space:
  ece_cache_total_size{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="ActiveSession", unit="mb"} 1.73
  ece_cache_total_size{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="Balance", unit="mb"} 0.52
  ece_cache_total_size{nodeName="ecs1", service="BRMFederatedCache", machine="abc.us.oracle.com", cache="Customer", unit="mb"} 1.17

### SESSION_AND_EVENT_METRICS

The SESSION_AND_EVENT_METRICS group contains metrics on ECE server sessions and rated events processed by these sessions. See "BRS_CLIENT_METRICS" for the Diameter session metrics.

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_rated_events_current_by_node</td>
<td>Gauge</td>
<td>Contains the total number of RatedEvent objects extracted from the Oracle NoSQL database since the last query. <strong>Note:</strong> In single-schema systems, only one Rated Event Formatter instance is tracked for collecting metrics.</td>
</tr>
<tr>
<td>ece_rated_events_total_by_node</td>
<td>Counter</td>
<td>Contains the total number of rated events processed by each ECS node. This includes both the rated events that are stored in the RatedEvent cache and the Oracle NoSQL database. This data is tracked only when you run ECE Metrics Collector with the -mp parameter.</td>
</tr>
<tr>
<td>ece_rated_events_total</td>
<td>Counter</td>
<td>Tracks the processing of rated events by ECE server nodes.</td>
</tr>
<tr>
<td>ece_session_metrics</td>
<td>Counter</td>
<td>Contains the total number of sessions opened or closed on each ECE server node by rating group. This metric can be refined when you run ECE Metrics Collector with the -mp parameter.</td>
</tr>
</tbody>
</table>

**Example**

The following are the examples for the metrics in the SESSION_AND_EVENT_METRICS group:
- **ece\_rated\_events\_current\_by\_node**. This example shows that 4477 events were extracted in the last scrape interval:

```
ece_rated_events_current_by_node(nodeName="ratedEventFormatter1", machine="abc.us.oracle.com", role="ratedEventFormatter", type="extracted") 4477.0
```

- **ece\_rated\_events\_total\_by\_node**. This example shows the rated events extracted and processed by ECS node **ecs1**. There are 158421 events stored in the RatedEvent cache, of which 158290 have been stored in the Oracle NoSQL database:

```
ece_rated_events_total_by_node(nodeName="ecs3", machine="abc.us.oracle.com", role="server", type="inserted") 177644.0
ece_rated_events_total_by_node(nodeName="ecs3", machine="abc.us.oracle.com", role="server", type="pushed") 177502.0
ece_rated_events_total_by_node(nodeName="ratedEventFormatter1", machine="abc.us.oracle.com", role="ratedEventFormatter", type="extracted") 503713.0
```

- **ece\_rated\_events\_total**. This example shows the number of rated events extracted from Rated Event Formatter and the number of rated events stored in the RatedEvent cache and Oracle NoSQL database respectively:

```
ece_rated_events_total(type="extracted") 503713.0
ece_rated_events_total(type="inserted") 511258.0
ece_rated_events_total(type="pushed") 510842.0
```

- **ece\_session\_metrics**. This example shows the number of sessions opened and closed for rating group 10 for the ECE server node **ecs3** (if **-mp** parameter is used):

```
ece_session_metrics(nodeName="ecs3", machine="abc.us.oracle.com", role="server", type="Close", rating_group="10") 174647.0
ece_session_metrics(nodeName="ecs3", machine="abc.us.oracle.com", role="server", type="Open", rating_group="10") 184504.0
ece_session_metrics(type="Close", rating_group="10") 506267.0
ece_session_metrics(type="Open", rating_group="10") 534832.0
```

**SERVICE\_METRICS**

The SERVICE\_METRICS group contains metrics for the Oracle Coherence cache services. You can use this data to monitor the server latency and load (per node) and the backlogs which may accumulate on the ECE nodes. You can configure the frequency to reset the Coherence service statistics by running the ECE Metrics Collector with the **-r <duration>** parameter.

<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_service_avg_thread_count</td>
<td>Gauge</td>
<td>Contains the average active thread count as determined by Coherence.</td>
</tr>
<tr>
<td>ece_service_endangered_partitions</td>
<td>Gauge</td>
<td>Contains the total number of endangered partitions on an ECE service. The metric value is 0 if the partition is not endangered and a number greater than zero if any node has failed.</td>
</tr>
<tr>
<td>Metric Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ece_service_ha_status</td>
<td>Gauge</td>
<td>Contains a numeric representation of the Coherence high-availability status. The metric value is 0 for ENDANGERED node, 1 for NODE-SAFE, 2 for RACK-SAFE, 3 for MACHINE-SAFE, and 4 for SITE-SAFE. Typically, nodes can be endangerd temporarily during rebalancing operations when the nodes are added or removed.</td>
</tr>
<tr>
<td>ece_service_request_avg_duration</td>
<td>Gauge</td>
<td>Contains the average server-side request latency in milliseconds.</td>
</tr>
<tr>
<td>ece_service_request_count</td>
<td>Counter</td>
<td>Contains the total number of requests executed by the service since the last reset.</td>
</tr>
<tr>
<td>ece_service_request_max_duration</td>
<td>Gauge</td>
<td>Contains the maximum server-side request latency (in milliseconds) since the last reset.</td>
</tr>
<tr>
<td>ece_service_request_pending_count</td>
<td>Gauge</td>
<td>Contains the total number of requests currently pending for a service. Large number of pending tasks may indicate a performance or capacity problem.</td>
</tr>
<tr>
<td>ece_service_request_pending_duration</td>
<td>Gauge</td>
<td>Contains the duration of the request (in milliseconds) pending in a service.</td>
</tr>
<tr>
<td>ece_service_task_avg_duration</td>
<td>Gauge</td>
<td>Contains the average server-side task latency in milliseconds since the last reset.</td>
</tr>
<tr>
<td>ece_service_task_backlog</td>
<td>Gauge</td>
<td>Contains the current server-side task backlog for each service. A large backlog is indicative of a performance or capacity problem.</td>
</tr>
<tr>
<td>ece_service_task_count</td>
<td>Counter</td>
<td>Contains the total number of tasks processed by a service since the last reset.</td>
</tr>
<tr>
<td>ece_service_task_max_backlog</td>
<td>Gauge</td>
<td>Contains the maximum task backlog for each service since the last reset.</td>
</tr>
<tr>
<td>ece_service_unbalanced_partitions</td>
<td>Gauge</td>
<td>Contains the total number of unbalanced partitions for a particular service. Typically, unbalanced partitions can occur temporarily during rebalancing operations when the nodes are added or removed.</td>
</tr>
</tbody>
</table>

**Example**

The following are the examples for the metrics in the SERVICE_METRICS group:

- **ece_service_avg_thread_count**. This example shows the number of BRMFederatedCache and InvocationService service threads in use. By default, ECE provides 4 threads for the BRMFederatedCache service and 33 threads for the InvocationService:

  ```
  ece_service_avg_thread_count{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 0.5684553
  ece_service_avg_thread_count{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 1.3715283
  ```

- **ece_service_endangered_partitions**. This example shows that there are no endangered partitions for the ECS server node ecs1:
ece_service_endangered_partitions{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 0
ece_service_endangered_partitions{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} -1

- **ece_service_ha_status.** This example shows that the BRMFederatedCache services on the ECS server nodes ecs1 and ecs2 are 1 (NODE-SAFE):

  ece_service_ha_status{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 1
  ece_service_ha_status{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} -1
  ece_service_ha_status{nodeName="ecs2", machine="abc.us.oracle.com", service="BRMFederatedCache"} 1
  ece_service_ha_status{nodeName="ecs2", machine="abc.us.oracle.com", service="InvocationService"} -1

- **ece_service_request_avg_duration.** This example shows the ECE server node ecs1 with average request for the BRMFederatedCache service as 5 milliseconds and 0 for the InvocationService service (this service processes only tasks):

  ece_service_request_avg_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 5.173
  ece_service_request_avg_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 0.000

- **ece_service_request_count.** This example shows the ECS server node ecs1 has processed approximately 30000 requests since the last statistics reset:

  ece_service_request_count{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 30044
  ece_service_request_count{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 0

- **ece_service_request_max_duration.** This example shows the maximum request duration on the ECE server node ecs1 is approximately 66 milliseconds:

  ece_service_request_max_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 67
  ece_service_request_max_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 0

- **ece_service_request_pending_count.** This example shows the ECE server node ecs1 with 12 pending requests:

  ece_service_request_pending_count{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 12
  ece_service_request_pending_count{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 0

- **ece_service_request_pending_duration.** This example shows the ECE server node ecs1 with requests pending for 13 milliseconds:

  ece_service_request_pending_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 13
  ece_service_request_pending_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 0

- **ece_service_task_avg_duration.** This example shows the average task durations on the ECE server node ecs1 for the InvocationService and BRMFederatedCache services in milliseconds:

  ece_service_task_avg_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 3.276
ece_service_task_avg_duration{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 9.142

- **ece_service_task_backlog.** This example shows the ECE server node ecs1 with a backlog of 4 requests on the BRMFederatedCache service and 1 task on the InvocationService on ECE server node ecs2.

  ece_service_task_backlog{nodeName="ecs1", machine="slc15ejg.us.oracle.com", service="BRMFederatedCache"} 4
  ece_service_task_backlog{nodeName="ecs1", machine="slc15ejg.us.oracle.com", service="InvocationService"} 0
  ece_service_task_backlog{nodeName="ecs2", machine="slc15ejg.us.oracle.com", service="BRMFederatedCache"} 0
  ece_service_task_backlog{nodeName="ecs2", machine="slc15ejg.us.oracle.com", service="InvocationService"} 1

- **ece_service_task_count.** This example shows that all services have handled approximately 15000 tasks:

  ece_service_task_count{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 14995
  ece_service_task_count{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 15010
  ece_service_task_count{nodeName="ecs2", machine="abc.us.oracle.com", service="BRMFederatedCache"} 14272
  ece_service_task_count{nodeName="ecs2", machine="abc.us.oracle.com", service="InvocationService"} 14264

- **ece_service_task_max_backlog.** This example shows the ECE server node ecs1 and ecs2 with a backlog of 26 and 13 tasks for the BRMFederatedCache and 0 and 1 for the InvocationService service:

  ece_service_task_max_backlog{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 26
  ece_service_task_max_backlog{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} 0
  ece_service_task_max_backlog{nodeName="ecs2", machine="abc.us.oracle.com", service="BRMFederatedCache"} 13
  ece_service_task_max_backlog{nodeName="ecs2", machine="abc.us.oracle.com", service="InvocationService"} 1

- **ece_service_unbalanced_partitions.** This example shows that there are no unbalanced partitions on ECE server node ecs1:

  ece_service_unbalanced_partitions{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache"} 0
  ece_service_unbalanced_partitions{nodeName="ecs1", machine="abc.us.oracle.com", service="InvocationService"} -1

FEDERATION_METRICS

The FEDERATION_METRICS group contains metrics for the ECE federated caches when ECE persistence is disabled. The metrics in this group provide information regarding the volume of data transferred, the number of objects transferred, and so on. You can use this metric to monitor the data transferred from the primary production system to the remote or backup systems. This data is typically used for disaster recovery where the Oracle NoSQL database is used for storing rated events.

You can run the ECE Metrics Collector with the `-f ClusterName` and `-F ServiceName` parameters to retrieve the federated cache metrics.
<table>
<thead>
<tr>
<th>Metric Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ece_federated_service_bandwidth</td>
<td>Gauge</td>
<td>Tracks the current or maximum bandwidth used to transfer data from ECE charging nodes to a secondary ECE cluster.</td>
</tr>
<tr>
<td>ece_federated_service_rate</td>
<td>Gauge</td>
<td>Tracks the approximate rate of data transfer in bytes and number of messages sent. This metric uses the Coherence Mbean attributes for tracking data.</td>
</tr>
<tr>
<td>ece_federated_service_replicate_millis</td>
<td>Gauge</td>
<td>Contains the cache replication latency (in milliseconds) for initial data replication. The type can be: total (total time taken for replication), estimate_ttc (estimated time to complete).</td>
</tr>
<tr>
<td>ece_federated_service_replicate_percent</td>
<td>Gauge</td>
<td>Contains the percentage of cache replication completed for a service.</td>
</tr>
<tr>
<td>ece_federated_service_status</td>
<td>Gauge</td>
<td>Contains the status of the service which is on federation. The state of a service can be: 1 (Initial), 2 (Idle), 3 Ready, 4 (Sending), 5 (Connecting), 6 (Connect_Wait), 7 (Stopped), 8 (Paused), 9 (Error), 10 (Yielding), 11 (Backlog_Excessive), 12 (Backlog_Normal), and 13 (Disconnected). The status of a service can be: 1 (OK), 2 (Warning), and 3 (Error).</td>
</tr>
<tr>
<td>ece_federated_service_time_millis</td>
<td>Gauge</td>
<td>Contains the cache replication latency (in milliseconds) for data replicated to the remote cache. The type can be: apply, backlog_delay, and round_trip. These are the 90th percentile latency times.</td>
</tr>
<tr>
<td>ece_federated_service_total</td>
<td>Gauge</td>
<td>Contains the total number of bytes, cache entries, and messages that are replicated to the remote cache. The entity type can be: records (total number of journal records), bytes (total number of bytes sent), entries (total number of cache entries sent), message (total number of messages sent), response (total number of message responses received). The status can be: sent (entity shipped to remote cluster), unacked (messages sent without acknowledgment), and error (messages failed).</td>
</tr>
</tbody>
</table>

**Example**

The following are the examples for the metrics in the FEDERATION_METRICS group:

- **ece_federated_service_bandwidth.** This example shows the bandwidth at the current moment for the BRMFederatedCache and ReplicatedFedratedCache services on the ECE charging server node ecs1 (in MBps). It also shows that there is no maximum bandwidth configured (as the value is -1.0) for these services:

  ece_federated_service_bandwidth{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache", type="current"} 9.159515380859375
  ece_federated_service_bandwidth{nodeName="ecs1", machine="abc.us.oracle.com", service="ReplicatedFedratedCache", type="current"} -1.0
**ece_federated_service_bandwidth**

- **service=BRMFederatedCache**, **type=max** -1.0
- **service=ReplicatedFederatedCache**, **type=current** 0.0
- **service=ReplicatedFederatedCache**, **type=max** -1.0

**ece_federated_service_rate**. This example shows the approximate number of bytes sent per second and number of messages sent for the BRMFederatedCache service on the ECE charging server node ecs1:

- **service=BRMFederatedCache**, **entity=bytes**, **type=sent** 1183714
- **service=BRMFederatedCache**, **entity=message**, **type=sent** 182

**ece_federated_service_replicate_millis**. This example shows the replication completed time in milliseconds for the BRMFederatedCache service on the ECE charging server node ecs1:

- **service=BRMFederatedCache**, **type=estimate_ttc** 0
- **service=BRMFederatedCache**, **type=total** 378
- **service=ReplicatedFederatedCache**, **type=estimate_ttc** 0
- **service=ReplicatedFederatedCache**, **type=total** 938

**ece_federated_service_replicate_percent**. This example shows that the replication is 100% complete for BRMFederatedCache and ReplicatedFederatedCache services on the ECE charging server node ecs1:

- **service=BRMFederatedCache** 100
- **service=ReplicatedFederatedCache** 100

**ece_federated_service_status**. This example shows the state and statuses of the BRMFederatedCache and ReplicatedFederatedCache services on the ECE charging server node ecs1. The service status is directly reflected from Coherence.

- **service=BRMFederatedCache**, **type=state** 2
- **service=ReplicatedFederatedCache**, **type=state** 2

**ece_federated_service_time_millis**. This example shows the time taken to apply the changes to the BRMFederatedCache service on the ECE charging server node ecs1, with a round-trip time of approximately 5 minutes:

- **service=BRMFederatedCache**, **type=apply** 20
- **service=BRMFederatedCache**, **type=backlog_delay** 10
• **ece_federated_service_total.** This example shows the total number of BRMFederatedCache service entries, messages, data bytes, records, and responses replicated from the ECE charging server node ecs1. The changes to the cache entries are grouped into journal records and then into messages:

```plaintext
ece_federated_service_total{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache", entity="bytes", type="sent"} 28868414
ece_federated_service_total{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache", entity="entries", type="sent"} 11292
ece_federated_service_total{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache", entity="message", type="sent"} 5590
ece_federated_service_total{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache", entity="message", type="unacked"} 0
ece_federated_service_total{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache", entity="records", type="sent"} 5646
ece_federated_service_total{nodeName="ecs1", machine="abc.us.oracle.com", service="BRMFederatedCache", entity="response", type="error"} 0
```

### Setting Log Levels by Using Scripts

You can set the log level of the ECE module or modules by ECE functional domain.

To set log levels by using the **ece_log** script:

1. On the driver machine, change to the `ECE_home/tools` directory.
2. Start ECC, if not started:
   ```bash
   ./ecc
   ```
3. Run the following command:
   ```bash
   ece_log -l Levels -c Classes -S Level
   ```
   where:
   - **Levels** specifies the log-levels to be selected; for example, "debug|info".
   - **Classes** specifies the log class names.
   - **Level** specifies the log-level to be set.
   
   For example:
   ```bash
   ece_log -l debug -c "rating|client" -S error
   ```
   This command sets all the DEBUG-level log entries to ERROR for the rating and client classes.

### Using cohql_file_processor for Query

The **cohql_file_processor** tool is a wrapper around the query tool (**query.sh**), which executes queries on ECE Coherence caches. You can use the following scripts to query ECE data instead of running the **query.sh** script frequently:

- **start_cohql_file_processor.** Use this script to run the query in the non-interactive mode using input files.
- **interactive_cohql.** Use this script to run the query in the interactive mode using command-line.

You can stop the query by using the **stop_cohql_file_processor** script.
Duplicate Check Enhancement in ECE

ECE performs duplicate check on the incoming TERMINATE, DIRECT_DEBIT, and REFUND requests to verify the session ID in the requests. By default, the session ID is stored in the aggregate usage and rated event objects. In this case, sometimes the aggregate usage and rated event objects have to be kept in the ECE cache for a longer duration when duplicate checks are performed on these requests.

With this enhancement, when data persistence is enabled in ECE, the session IDs are stored in the new Session ID cache, `TerminatedSessionHistory`. This cache stores only the session IDs of the requests that are processed. For performing duplicate checks on TERMINATE, DIRECT_DEBIT, and REFUND requests, ECE checks the session IDs in this new cache. This ensures that the aggregate usage and rated event objects are evicted from the cache immediately after the successful cache update.

Charging Operation Type Can be Configured for Expired Active Session Cleanup

In the previous releases, ECE removed the expired active session and the corresponding expired balance reservations only when TERMINATE or CANCEL operation type in usage requests were processed. It was not possible to configure other charging operation types for cleaning up the expired objects.

With this enhancement, you can configure for which charging operation types the expired active sessions and balance reservations must be cleaned up. For example, you can configure ECE to remove the expired active sessions and balance reservations when the INITIATE or UPDATE operation type in a usage request is processed. You can specify the charging operation types to be used by setting the `asoCleanupOperationList` Mbean attribute. See "Configuring Expired Objects Cleanup".

Configuring Expired Objects Cleanup

1. Access the ECE MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Start a JMX editor, such as JConsole, that enables you to edit MBean attributes.
   d. Connect to the ECE charging server node set to `start CohMgt = true` in the `ECE_home/locceceserver/config/eceTopology.conf` file.

      The `eceTopology.conf` file also contains the host name and port number for the node.

   e. In the editor’s MBean hierarchy, expand the `ECE Configuration` node.

2. Expand `charging.server`.

3. Expand `Attributes`.
4. Set the `asoCleanupOperationList` attribute to the charging operation types for which the expired active session and balance reservation objects must be cleaned up.

   The default values are TERMINATE, CANCEL. If this attribute is set to null, the default values are used for the cleanup.

   For the list of charging operation types supported by ECE, see the discussion about the Charging API in *ECE Implementing Charging*.

---

**ECE Now Generates POID for Events**

In the previous releases, ECE was using the Portal object IDs (POIDs) received from BRM for tracking events rated by ECE.

With this enhancement, POIDs can be generated in ECE for tracking the rated events. ECE uses Rated Event Formatter to generate the required POIDs and persists the last allocated POID ID in the Oracle NoSQL database. This ensures that the POIDs are generated without any duplication even if the ECE system is restarted.

The POID generated in ECE contains the following information:

```
event_type date cluster_id BRM_schema_id unique_id
```

See Table 3-1 for the description of each entry in the POID.

### Table 3-1  POID Entries in ECE

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>event_type</code></td>
<td>A unique 4-bit number assigned to each event type. For example, 0 is assigned to subscription events, 1 is assigned to postpaid events (USAGE_POSTPAID), and 2 to 7 is assigned to prepaid events (USAGE_PREPAID) depending on the <code>prepaidPartitionSet</code> value specified in BRM. The default value for <code>event_type</code> is 0.</td>
</tr>
<tr>
<td><code>date</code></td>
<td>The 16-bit date on which the POID is generated. The date is determined based on ECE <code>virtualTime</code> if it is enabled. For more information on <code>virtualTime</code>, see the discussion about changing time and date to test ECE in <em>BRM Elastic Charging Engine Implementing Charging</em>.</td>
</tr>
<tr>
<td><code>cluster_id</code></td>
<td>A unique 4-bit number assigned to the Coherence cluster to identify ECE in the cluster. The <code>cluster_id</code> is limited to 0 to 15 and the maximum number of ECE clusters allowed in a deployment is 16. The default value for <code>cluster_id</code> is 0. If ECE is configured for disaster recovery, you must specify the cluster ID for each cluster used in the Active-hot standby or Active-cold standby systems.</td>
</tr>
<tr>
<td><code>BRM_schema_id</code></td>
<td>A unique 6-bit number assigned to the BRM schema. The <code>BRM_schema_id</code> is limited to 0 to 31.</td>
</tr>
<tr>
<td><code>unique_id</code></td>
<td>A unique 34-bit number assigned to each POID.</td>
</tr>
</tbody>
</table>

For tracking the events rated by ECE, Rated Event Formatter uses the POIDs generated in ECE. You can configure multiple instances of Rated Event Formatter to ensure high availability and uninterrupted POID allocation. In case if the primary Rated Event Formatter instance fails, the secondary Rated Event Formatter instance ensures
that the POIDs are allocated without any interruption. In a disaster recovery deployment, if the Rated Event Formatter instance in the primary site fails, the Rated Event Formatter instance in the backup site continues the POID allocation for the events. To connect the instances in different sites or systems, you must specify the name of the primary Rated Event Formatter instance in the primary and secondary Rated Event Formatter instances.

For tracking the bill items created in ECE, ECE continues to use the POIDs received from BRM. However, ECE now persists the POID pool received from BRM in the Oracle NoSQL database. This ensures that the reserved POID pool is retained in ECE even after the ECE restart. It allows ECE to continue the POID allocation for the bill items using the existing POID pool, which in turn reduces the dependency on BRM.

To enable POID generation in ECE for events, you must perform the following:

1. Enable prepaid-event partitions in BRM. For instructions, see "Enabling Prepaid-Event Partitions".
2. Ensure that the cluster ID is configured for ECE clusters. The cluster ID must be specified if you have ECE configured for disaster recovery. See "Configuring Cluster ID".
3. Ensure that the name of the primary Rated Event Formatter instance is specified in each Rated Event Formatter instance. The primary Rated Event Formatter instance must be specified if you have ECE configured for disaster recovery. See "Connecting Rated Event Formatter Instances".
4. Enable prepaid-event partitions in ECE. See "Enabling Prepaid-Event Partitions".

### Configuring Cluster ID

To configure the cluster ID for ECE clusters:

1. Access the ECE MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Start a JMX editor, such as JConsole, that enables you to edit MBean attributes.
   d. Connect to the ECE charging server node set to start CohMgt = true in the ECE_home/oceserver/config/ecetopology.conf file.

      The ecetopology.conf file also contains the host name and port number for the node.

   e. In the editor's MBean hierarchy, expand the ECE Configuration node.

2. Expand charging.clusters.Cluster_Name, where Cluster_Name is the name of the ECE cluster that you are configuring.

3. Expand Attributes.

4. Set the id attribute to a unique number that indicates the ID of the cluster in the POID generated in ECE.

   Rated Event Formatter uses the cluster ID in the POID to identify the ECE clusters. The cluster ID must be unique for each cluster.
Connecting Rated Event Formatter Instances

To connect the Rated Event Formatter instances in different sites or systems, you must perform this for each Rated Event Formatter instance

To connect Rated Event Formatter instances:

1. Access the ECE MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Start a JMX editor, such as JConsole, that enables you to edit MBean attributes.
   d. Connect to the ECE charging server node set to \texttt{start CohMgt = true} in the \texttt{ECE\_home\locceserver\config\eceTopology.conf} file.
      The \texttt{eceTopology.conf} file also contains the host name and port number for the node.
   e. In the editor's MBean hierarchy, expand the \texttt{ECE Configuration} node.

2. Expand \texttt{charging.ratedEventFormatters.\textit{Instance\_Name}}, where \textit{Instance\_Name} is the name of the instance you want to configure; for example, \texttt{ratedEventFormatter2}.

3. Expand Attributes.

4. Set the \texttt{primaryInstanceName} attribute to the name of the primary Rated Event Formatter instance.
   For example, if the name of the primary Rated Event Formatter instance is \texttt{ratedEventFormatter1}, specify \texttt{ratedEventFormatter1} as \texttt{primaryInstanceName} in the primary and all secondary instances.

5. Change directory to the \texttt{ECE\_home\locceserver\bin} directory.

6. Start ECC:
   \texttt{./ecc}

7. Stop and restart any Rated Event Formatter instances that you configured.
   Each instance reads its configuration information by name.
   For information about stopping and starting Rated Event Formatter instances, see the discussion about starting and stopping ECE in \textit{BRM Elastic Charging Engine System Administrator's Guide}.

Enabling Prepaid-Event Partitions

To enable prepaid-event partitions:

1. Access the ECE MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Start a JMX editor, such as JConsole, that enables you to edit MBean attributes.
d. Connect to the ECE charging server node set to `start CohMgt = true` in the `ECE_home/occeserver/config/eceTopology.conf` file.
   
The `eceTopology.conf` file also contains the host name and port number for the node.

e. In the editor’s MBean hierarchy, expand the `ECE Configuration` node.

1. Expand `charging.brmCdrPlugins.Instance_Name`, where `Instance_Name` is the name of the BrmCdrPluginDirect Plug-in instance you are configuring.

2. Expand `Attributes`.

3. Set the `prepaidPartitionSet` attribute to the value that you specified in the `prepaid_partition_set` entry in the `BRM_Home/sys/dm_oracle/pin.conf` file.

   To enable prepaid-event partitions, you need to set this attribute to a number between 2 and 7. If this attribute is set to 0, ECE continues to use the POIDs received from BRM for events instead of generating them.

New Tool for Querying ECE Cache Data

In previous releases, the ECE cache data query took a long time to complete if the data was not properly indexed.

With this enhancement, a new query tool, `query_cache.sh`, has been introduced to query data in the ECE caches that are associated with customer keys. The new query tool provides access to the ECE cache content by way of CohQL. This tool is included with the ECE Server software in `ECE_home/occesdk/bin/query`, where `ECE_home` is the directory in which ECE Server software is installed. You can use this tool for debugging, development, generating reports, and other-related queries. This tool supports only the non-interactive mode.

The following is the syntax for the non-interactive use of the new query tool:

```
sh query_cache.sh run cache_name Associated_key ID
```

where:

- `Cache_name` specifies the name of the cache you are querying.
- `Associated_key` specifies the customer ID (composite key) that is associated with the cache.
- `ID` specifies the unique identifier, such as balance ID, subscriber ID, and so on.

For example, to query a customer's balance, you can run the query on the Balance cache using the associated customer ID (479665838) and balance ID (479664046):

```
sh query_cache.sh run Balance 479665838 479664046
```

This query returns the balance information for the specified customer ID.

For information on the query tool options, use the help command:

```
sh query_cache.sh -h
```
ECE Now Supports Wildcard in Item Type Selectors

ECE now supports wildcard (*) in item type selectors for services and events. You can use the wildcard to substitute one or more characters in the service or event type to indicate that any value is acceptable; for example, /service/telco/gsm/*.

If wildcard is used in the service or event type, ECE uses the applicableToAllChildServices and applicableToAllChildEvents values to identify if the service or event type and item type selector is applicable for all the child services or events. If the value is true, the item type selector is considered for all the child services or events. If the value is false, the item type selector is not considered for the child services or events.

For more information on using wildcard in item type selectors, see PDC Creating Product Offerings.

Support for Persisting BRS Configuration Data

In the previous releases, batch request service (BRS) parameters were not persisted in the ECE cache and were lost during ECE restart. You had to manually set these parameters again after ECE restart.

With this enhancement, you can persist the BRS parameters in the ECE cache by adding them using the brsConfigurations MBean attribute. You can now use this attribute instead of the BatchRequestService and ChargingClient.BatchRequestService MBean attributes for configuring system overload protection and client-side ECE request queues.

Note:

If you have multiple instances of a component and you want to use the same BRS configuration for all the instances, you can add the BRS configuration for the role instead of each instance. For example, if you have diameterGateway1, diameterGateway2, and diameterGateway3, add the BRSconfiguration instance as diameterGateway to use the same BRS configuration for all the three instances.

If the BRS configuration is not added or removed for any of the instance, the BRS configuration for the role is applied to that instance by default. If the BRS configuration for the role is not defined, the default (system-level) BRS configuration is applied to that instance.

To add the BRS configuration:

1. Access the ECE MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Start a JMX editor, such as JConsole, that enables you to edit MBean attributes.
d. Connect to the ECE charging server node set to `start CohMgt = true` in the `ECE_home/locceceserver/config/eceTopology.conf` file.

The `eceTopology.conf` file also contains the host name and port number for the node.

e. In the editor’s MBean hierarchy, expand the **ECE Configuration** node.

2. Expand **charging.brsConfigurations**.
3. Expand **Operations**.
4. Click **addBrsConfiguration**.
5. Set the **name** attribute to the name of the BrsConfiguration instance; for example, `diameterGateway1`.
6. Click **addBrsConfiguration**.
   The new BRS configuration is added. For example, `charging.brsConfigurations.diameterGateway1`.
7. Expand **charging.brsConfigurationsinstancename**, where `instancename` is the name of the instance for which you are configuring BRS.
8. Expand **Attributes**.
9. Specify the values for the attributes.

For descriptions of each attribute, see the discussion about configuring system overload protection and configuring client-side ECE request queues in *BRM System Administrator's Guide*. For the default values, see the `charging.brsConfigurations.default` Mbean attribute.

You can also delete the BRS configuration. To delete the BRS configuration:

1. Access the ECE MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Start a JMX editor, such as JConsole, that enables you to edit MBean attributes.
   d. Connect to the ECE charging server node set to `start CohMgt = true` in the `ECE_home/locceceserver/config/eceTopology.conf` file.
      The `eceTopology.conf` file also contains the host name and port number for the node.
   e. In the editor’s MBean hierarchy, expand the **ECE Configuration** node.
2. Expand **charging.brsConfigurations**.
3. Expand **Operations**.
4. Click **removeBrsConfiguration**.
5. Set the **name** attribute to the name of the BrsConfiguration instance that you want to remove.
6. Click **removeBrsConfiguration**.
   The BRS Configuration is deleted.
Rated Event Partition in the ECE Persistent Database

In the previous releases, Rated Event Formatter was using SQL-DELETE query for deleting the rated events stored in the ECE persistence database. When SQL-DELETE query was used for deleting rated events, there was a considerable increase in the table space. This might impact the overall performance.

With this enhancement, Rated Event Formatter uses Oracle Drop Partition instead of SQL-DELETE query for deleting rated events. By default, the partition time for creating rated event partitions is set to 5 minutes. You can change the default partition time depending on your sizing and performance requirements by setting the `ratedEventTablePartitionByMinute` MBean attribute. During installation you can configure the initial storage and subpartition values for rated event table. For more information, see *Elastic Charging Engine Installation Guide*.

To change the partition time:

1. Access the ECE MBeans:
   a. Log on to the driver machine.
   b. Start the ECE charging servers (if they are not started).
   c. Start a JMX editor, such as JConsole, that enables you to edit MBean attributes.
   d. Connect to the ECE charging server node set to `start CohMgt = true` in the `ECE_home/oceceserver/config/eceTopology.conf` file.
      
      The `eceTopology.conf` file also contains the host name and port number for the node.
   e. In the editor's MBean hierarchy, expand the ECE Configuration node.

2. Expand `charging.connectionConfigurations.Connection_Name`; where `Connection_Name` is the name of the persistence database connection.

3. Expand Attributes.

4. Click `ratedEventTablePartitionByMinute` to set the partition time.

5. Save your changes.

New Features in ECE 12.0/12.0 Patch Set 1

For the new features in ECE 12.0/12.0 Patch Set 1 (12.0.0.1.0), see *ECE Release Notes*. 
New Features in PDC

This chapter provides an overview of the new features in Oracle Communications Pricing Design Center (PDC) 12.0 Patch Set 1 and Patch Set 2.

New Features in PDC 12.0 Patch Set 2

PDC 12.0 Patch Set 2 includes the following enhancements:

- Support for Configuring Item Type Selectors in the PDC User Interface
- PDC Web Services for Creating and Exporting Components
- PDC Synchronizes Event Data Using Event Types
- PDC Now Supports Wildcard in Item Type Selectors
- Support for Enabling SSO Using SAML

Support for Configuring Item Type Selectors in the PDC User Interface

In previous releases, you could configure item type selector rules only by importing them into the PDC database by using the `ImportExportPricing` utility.

You can now configure item type selector rules by using the PDC user interface. For more information, see the discussion about configuring the item type selectors in the PDC Online Help.

PDC Web Services for Creating and Exporting Components

In previous releases, you could use PDC Web services only to create and modify pricing components. With this enhancement, you can also use PDC Web services to perform the following:

- Export operation for Pricing components.
- Create, modify, and export operations for Setup components.
- Create, and export operations for Metadata, and Custom data.

Creating or Modifying Components using PDC Web Services

The PDC Web service uses the `PricingGateway.xsd` file for creating or modifying components in PDC. The XSD files describe the structure of the XML document. The XML file you create must comply with the structure defined in the XSD. The `PricingGateway.xsd` file is available at:

http://hostName:sslPortNumber/pdc/PricingGatewayPort?xsd=1

where:

- `hostName` is the host name of the machine on which PDC is deployed.
- **sslPortNumber** is the SSL port number of the domain on which PDC is deployed.

You can use the PDC Web service to do the following:

- Create the setup components defined in an XML file in PDC. See "createBusinessConfig" for more information.
- Create the setup components defined in an XML file in PDC and publish the setup components to a billing system, such as Oracle Communications Billing and Revenue Management (BRM). See "createBusinessConfigAndSubmit" for more information.
- Modify the setup components in PDC as defined in an XML file. See "modifyBusinessConfig" for more information.
- Modify the setup components in PDC as defined in an XML file and publish the setup components to a billing system. See "modifyBusinessConfigAndSubmit" for more information.
- Create the metadata defined in an XML file in PDC. See "createMetadata" for more information.
- Create the custom data defined in an XML file in PDC. See "createCustomFields" for more information.

### createBusinessConfig

This Web service operation validates the input XML by comparing the XML fields and values against the values in the **PricingGateway.xsd** file and the rules for each type of Business config component. If the validation is successful, it retrieves the data from the XML file and creates Business config components in PDC.

The **createBusinessConfig** operation does not publish the Business config components to the BRM database.

**Syntax**

```java
public oracle.communications.brm.pdc.server.service.types.PDCResponseType createBusinessConfig(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws oracle.communications.brm.pdc.server.service.PricingExceptionResponse;
```

### createBusinessConfigAndSubmit

This Web service operation validates the input XML by comparing the XML fields and values against the values in the **PricingGateway.xsd** file and the rules for each type of Business config component. If the validation is successful, it retrieves the data from the XML file, creates Business config components in PDC, and publishes the Business config components to the BRM database.

**Syntax**

```java
public oracle.communications.brm.pdc.server.service.types.PDCResponseType createBusinessConfigAndSubmit(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws oracle.communications.brm.pdc.server.service.PricingExceptionResponse;
```
modifyBusinessConfig

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of Business config component. If the validation is successful, it retrieves the data from the XML file and updates the existing Business config components in PDC.

The modifyBusinessConfig operation does not publish the Business config components to the BRM database.

Syntax:

```java
public oracle.communications.brm.pdc.server.service.types.PDCResponseType
modifyBusinessConfig(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws
oracle.communications.brm.pdc.server.service.PricingExceptionResponse;
```

modifyBusinessConfigAndSubmit

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of Business config component. If the validation is successful, it retrieves the data from the XML file and updates the existing Business config components in PDC, and publishes the Business config components to the BRM database.

Syntax:

```java
public oracle.communications.brm.pdc.server.service.types.PDCResponseType
modifyBusinessConfigAndSubmit(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws
oracle.communications.brm.pdc.server.service.PricingExceptionResponse;
```

createMetadata

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of Metadata component. If the validation is successful, it retrieves the data from the XML file and creates Metadata components in PDC.

The createMetadata operation does not publish the Metadata components to the BRM database.

Syntax:

```java
public oracle.communications.brm.pdc.server.service.types.PDCResponseType
createMetadata(oracle.communications.brm.pdc.server.service.types.PricingInputXMLType param) throws
oracle.communications.brm.pdc.server.service.PricingExceptionResponse;
```

createCustomFields

This Web service operation validates the input XML by comparing the XML fields and values against the values in the PricingGateway.xsd file and the rules for each type of Custom Fields. If the validation is successful, it retrieves the data from the XML file and creates Custom Fields in PDC.

The createCustomFields operation does not publish the Custom Fields to the BRM database.
Symantec

public oracle.communications.brm.pdc.server.service.types.PDCResponseType createCustomFields(oracle.communications.brm.pdc.server.service.types.PricingInputXML Type param) throws
oracle.communications.brm.pdc.server.service.PricingExceptionResponse;

getPricingObjects

This Web service operation validates the input XML by comparing the XML fields and values against the values in the RetrievePricingGateway.xsd file. If the validation is successful, it retrieves the Pricing Objects from PDC and returns the data in RetrieveResponseType object.

Syntax

public oracle.communications.brm.pdc.server.service.types.RetrieveResponseType getPricingObjects(oracle.communications.brm.pdc.server.service.types.RetrieveInputXML Type param) throws Exception;

getBusinessConfigObjects

This Web service operation validates the input XML by comparing the XML fields and values against the values in the RetrievePricingGateway.xsd file. If the validation is successful, it retrieves the Business Config from PDC and returns the data in RetrieveResponseType object.

Syntax

public oracle.communications.brm.pdc.server.service.types.RetrieveResponseType getBusinessConfigObjects(oracle.communications.brm.pdc.server.service.types.RetrieveInputXMLType param) throws Exception;

getMetadataObjects

This Web service operation validates the input XML by comparing the XML fields and values against the values in the RetrievePricingGateway.xsd file. If the validation is successful, it retrieves the Metadata Object from PDC and returns the data in RetrieveResponseType object.

Syntax

public oracle.communications.brm.pdc.server.service.types.RetrieveResponseType getMetadataObjects(oracle.communications.brm.pdc.server.service.types.RetrieveInputXMLType param) throws Exception;

getBRMObjects

This Web service operation validates the input XML by comparing the XML fields and values against the values in the RetrievePricingGateway.xsd file. If the validation is successful, it retrieves the BRM Object from PDC and returns the data in RetrieveResponseType object.
Syntax

public oracle.communications.brm.pdc.server.service.types.RetrieveResponseType
getBRMObjects(oracle.communications.brm.pdc.server.service.types.RetrieveInputXMLType
param) throws Exception;

getCustomFields

This Web service operation validates the input XML by comparing the XML fields and
values against the values in the RetrievePricingGateway.xsd file. If the validation is
successful, it retrieves the Custom Fields from PDC and returns the data in
RetrieveResponseType object.

Syntax

public oracle.communications.brm.pdc.server.service.types.RetrieveResponseType
getCustomFields(oracle.communications.brm.pdc.server.service.types.RetrieveInputXMLType
param) throws Exception;

Exporting Components using PDC Web Services

You can use the PDC Web services to do the following:

- Export the pricing components. See "getPricingObjects" for more information.
- Export the setup components. See "getBusinessConfigObjects" for more
  information.
- Export the metadata components. See "getMetadataObjects" for more information.
- Export the BRM components. See "getBRMObjects" for more information.
- Export custom fields. See "getCustomFields" for more information.

PDC Web services use the RetrieveInputXMLType object as input for exporting the
PDC components. You can create the RetrieveInputXMLType object.

- Using the Setter Method. See "Creating RetrieveInputXMLType Object Using
  Setter Method".
- Using the XML. See "Creating RetrieveInputXMLType Object Using XML".

Creating RetrieveInputXMLType Object Using Setter Method

You can create the RetrieveInputXMLType object using setter methods (for example,
pojos) with the required parameters to export components from PDC system.

For example, to export pricing components you can create the object as follows:

getPricingObjects() {
    RetrieveResponseType response = new RetrieveResponseType();
    RetrieveInputXMLType request = new RetrieveInputXMLType();
    ObjectType objectType = new ObjectType();
    WSPricingObject pricingObject = new WSPricingObject();
    pricingObject.getPricingObjectType().add(WSPricingObjectType.ALTERATION_EXCLUSION);
    objectType.setPricingObject(pricingObject);
    request.setObjectType(objectType);
    request.setAllReferences(false);
    request.setIncludeFailedObjects(false);
    request.setObsolete(false);
    request.setReferences(false);
Creating RetrieveInputXMLType Object Using XML

The PDC Web services use the `RetrievePricingGateway.xsd` file for exporting components from PDC. The XSD file describes the structure of the XML file. The XML file you create must comply with the structure defined in the XSD. The `RetrievePricingGateway.xsd` file is available at:

http://hostName:sslPortNumber/pdc/PricingGatewayPort?xsd=2

where:

- `hostName` is the host name of the machine on which PDC is deployed.
- `sslPortNumber` is the SSL port number of the domain on which PDC is deployed.

You need to manually set the required parameters in the XML file for indicating the objects that you want to export from the PDC system. The `RetrieveInputXMLType` object is then created by parsing this XML.

Parameters for Exporting PDC Components

Table 4-1 provides the description of parameters used for exporting PDC components.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>objectType</td>
<td>Specifies the object type to export. The valid values are <code>pricingObject</code>, <code>businessConfigObject</code>, <code>brmObject</code>, and <code>metadataObject</code>. If the pricing object type is not specified, all the pricing components are considered for exporting. This parameter is also applicable for setup components, metadata, and cross-reference data.</td>
</tr>
<tr>
<td>references</td>
<td>Specifies whether to export data including references.</td>
</tr>
<tr>
<td>allReferences</td>
<td>Specifies whether to export data including all references. This is applicable only when exporting pricing objects. When <code>allReferences</code> is set to <code>true</code> for pricing objects, the response is written separately into <code>pricingObjectResponse</code>, <code>configObjectResponse</code>, and <code>metadataObjectResponse</code> fields. If the export <code>allReferences</code> is not set for pricing objects, the pricing object responses are written into <code>response</code> field.</td>
</tr>
<tr>
<td>obsolete</td>
<td>Specifies whether to export obsolete data.</td>
</tr>
<tr>
<td>includeFailedObjects</td>
<td>Specifies whether to export data including failed objects.</td>
</tr>
</tbody>
</table>
Table 4-1  (Cont.) Parameters for Exporting PDC Components

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>modifiedAfterDate</td>
<td>Specifies whether to export data based on modified date.</td>
</tr>
<tr>
<td></td>
<td><strong>ModifiedAfterDate</strong> supports the following formats:</td>
</tr>
<tr>
<td></td>
<td>•  yyyy-mm-dd</td>
</tr>
<tr>
<td></td>
<td>Example: 2010-01-05</td>
</tr>
<tr>
<td></td>
<td>•  yyyy-mm-dd'T'hh:mm:ss</td>
</tr>
<tr>
<td></td>
<td>Example: 2010-01-05T23:59:59</td>
</tr>
<tr>
<td></td>
<td>•  yyyy-mm-dd'T'hh:mm:ssz</td>
</tr>
<tr>
<td></td>
<td>Example: 2010-01-05T19:05:09GMT+05:30</td>
</tr>
<tr>
<td>modifiedByUser</td>
<td>Specifies whether to export data based on the user who modified the data.</td>
</tr>
<tr>
<td>objectName</td>
<td>Specifies whether to export data based on object name.</td>
</tr>
<tr>
<td>UserContext</td>
<td>Specifies whether to export data for a specific user.</td>
</tr>
<tr>
<td>productSpecName</td>
<td>Specifies whether to export data based on product specification.</td>
</tr>
<tr>
<td></td>
<td>This is applicable only when exporting pricing components.</td>
</tr>
</tbody>
</table>

About Web Service Response

The PDC Web service operation first validates the input XML by comparing the XML fields and its values against the values in the RetrievePricingGateway.xsd file. If the validation is successful, it retrieves the objects from PDC and returns the data in RetrieveResponseType object.

Structure of the RetrieveResponseType object is as follows:

```xml
<xsd:complexType name="RetrieveResponseType">
  <xsd:documentation>Get Object Webservice Response</xsd:documentation>
  <xsd:sequence>
    <xsd:element name="status" type="xsd:string"/>
    <xsd:element name="errors" type="xsd:string" maxOccurs="unbounded" minOccurs="0"/>
    <xsd:element name="response" type="xsd:base64Binary"/>
    <xsd:element name="pricingObjectResponse" type="xsd:base64Binary"/>
    <xsd:element name="configObjectResponse" type="xsd:base64Binary"/>
    <xsd:element name="metadataObjectResponse" type="xsd:base64Binary"/>
  </xsd:sequence>
</xsd:complexType>
```

The RetrieveResponseType object contains the status of operation and the exported object fields in the XML binary format based on the input provided in RetrieveInputXml file. The status field value is either SUCCEEDED or FAILED. The RetrieveResponseType object also capture errors occurred during the operation.

Processing the Response from Export/GET API

When an export or GET operation is successful, you receive a response with the requested object data. You get the response as follows:

```java
response = pricingGatewayPortType.getPricingObjects(request);
```
PDC Synchronizes Event Data Using Event Types

With this enhancement, the BRM event data is synchronized with PDC based on the event type defined in BRM. For information on the list of event types in BRM, refer to "BRM Supports POID Generation in ECE".

PDC Synchronizes Event Data Using Event Types

With this enhancement, the BRM event data is synchronized with PDC based on the event type defined in BRM. For information on the list of event types in BRM, refer to "BRM Supports POID Generation in ECE".
**Note:**

When you set an event type in BRM, note the following:

- If the event type is set to **NONE** in BRM and the corresponding event is not mapped in the `pin_event_map` file, the event does not synchronize with PDC.
- If the event type is set to **NONE** in BRM and the corresponding event is mapped in the `pin_event_map` file, the event synchronizes with PDC. However, you should modify the event type in BRM from **NONE** to the relevant event type and then run the **SyncPDC** utility again if you want to use this event type for creating a pricing component.

After you create an event and synchronize it with PDC, it is not recommended to modify the event type of the event. However, if you have set the incorrect event type, you can modify the incorrect event type by updating it in BRM and then synchronizing in PDC. You can modify the event type only for the custom events. For example, if the event type for the parent event `/event/session/telco/gprs` is set as **USAGE_PREPAID** and the event type for the child event `/event/session/telco/gprs/master` is set as **NONE**, you must change the event type for the child event as **USAGE_PREPAID** in BRM, synchronize it with PDC, and then publish to ECE. See “Synchronizing and Publishing Event Type to ECE” for synchronizing the modified data to PDC and publishing it to ECE.

### Synchronizing and Publishing Event Type to ECE

To synchronize the event type with PDC and publish it to ECE:

1. Ensure that the event type is changed in BRM. For more information, see the discussion about changing the event type in “BRM Supports POID Generation in ECE”.
2. Run the **SyncPDC** utility.
3. Go to `PDC_home/apps/bin`, where `PDC_home` is the directory in which the PDC software is installed.
4. Export the event that you want to modify by using the following command:
   ```bash
   ImpotExportPricing -export filename.xml -metadata
   ```
   where `filename` is the name of the XML file.
5. Verify the exported data in the XML file and set the `<TRUE>` value to **TRUE** in the following syntax:
   ```xml
   <
   <1.0
   <TRUE>
   <
   ```
6. Import the modified XML file into PDC database by running the following command:
   ```bash
   ImpotExportPricing -import -metadata filename -ow
   ```
   Modified event type is published to ECE.
PDC Now Supports Wildcard in Item Type Selectors

PDC now supports wildcard (*) in item type selectors for services and events. You can use the wildcard to substitute one or more characters in the service or event name to indicate that any value is acceptable; for example, /service/telco/gsm*. The following elements are added in PDC to support wildcard: applicableToAllChildServices and applicableToAllChildEvents. By setting the true or false value to these elements, you can indicate whether the item type selector is applicable to child services or events.

If wildcard is used in the services and events, Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE), real-time rating engine, and batch rating engine use the applicableToAllChildServices and applicableToAllChildEvents values to identify if the services or events are applicable for all child services or events. If the value is set to true, the item type selector is considered for all the child services and events. If the value is set to false, the child services or events are not considered.

For more information, see the discussion about configuring item type selectors in PDC Creating Product Offerings.

Support for Enabling SSO Using SAML

You can now use SAML to enable single-sign on (SSO) in PDC. SSO allows you to log in to applications using a single user name and password combination.

For more information, see the discussion about configuring SAML for SSO in PDC Installation Guide.

New Features in PDC 12.0 Patch Set 1

PDC 12.0 Patch Set 1 includes the following enhancements:

• Support for Rolling Back the PDC Patch Set
• Support for Secure Communication With the PDC Database

Support for Rolling Back the PDC Patch Set

PDC now allows you to roll back a PDC patch set. For example, if you experience issues after installing PDC 12.0 Patch Set 1, you can roll back PDC to 12.0.

For more information, see the discussion about rolling back a patch set in PDC Installation Guide.

Support for Secure Communication With the PDC Database

PDC now supports Secure Sockets Layer (SSL)-enabled database to ensure secure communication between the PDC application and database.

For more information, see the discussions about configuring SSL for the Oracle database and configuring PDC to use SSL-enabled database in PDC Installation Guide.
5

New Features in Billing Care

This chapter provides an overview of the new features in Oracle Communications Billing Care 12.0 Patch Set 1 and Patch Set 2.

New Features in Billing Care 12.0 Patch Set 2

Billing Care 12.0 Patch Set 2 includes the following enhancements:

• SDK Enhancements for Customizing Billing Care
• View Bills Generated Before Moving the Account to a Hierarchy
• Support for Account Number and Bill Number in Billing Care Embeddable URLs
• Embeddable URLs for Purchase Package and Purchase Bundle Screens

For more information on these enhancements, see the Billing Care Online Help.

SDK Enhancements for Customizing Billing Care

You can now use the Billing Care SDK and OPSS policies to perform the following in Billing Care:

• Customize reason codes displayed for event adjustments.
• Limit the adjustment percentage entered by customer service representatives (CSRs) for event adjustments.
• Disable the event adjustment options based on user roles.
• Restrict debit and credit adjustment options for event adjustments.
• Make the Notes field mandatory for additional product purchase and event adjustments.
• Display only event adjustments in the Bills section for performing adjustments.
• Restrict the purchase of additional bundles based on user roles.
• Restrict the validity or end date set by the CSRs while purchasing additional products or services.
• Set the maximum adjustment limit based on the currency resources used for event adjustments.
• Filter the bundles available for purchase.
• Filter start and end dates for additional purchase.
• Disable the Charges not related to services link and the link to child accounts in the Bills tab, My Charges area, and Payment Details dialog box.

For more information, see Billing Care SDK Guide.
View Bills Generated Before Moving the Account to a Hierarchy

In previous releases, when an account was moved to a hierarchy as a child account, the bills generated for the account earlier were not displayed in the bills list.

With this enhancement, you can view the bills that are generated before and after moving the account to a hierarchy in the account's Bills > Switch Bills menu. For more information, see the discussion about switch bills in the Billing Care Online Help.

Support for Account Number and Bill Number in Billing Care Embeddable URLs

In the previous releases, only account ID and bill ID was supported in the Billing care embeddable URLs.

With this enhancement, you can also use account number and bill number instead of the corresponding POIDs in Billing Care URLs that are embedded in external applications.

For more information, see the discussion about embedding billing care screens in Billing Care SDK Guide.

Embeddable URLs for Purchase Package and Purchase Bundle Screens

Billing Care supports embedding screens into CRM applications and online account management interfaces to directly access the Billing Care functionality.

With this enhancement, you can embed Purchase Package and Purchase Bundle screens into external applications.

For more information, see the discussion about embedding billing care screens in Billing Care SDK Guide.

New Features in Billing Care 12.0 Patch Set 1

Billing Care 12.0 Patch Set 1 includes the following enhancements:

- Additional Options for Searching Bills
- Billing Care Is Now Certified with Mozilla Firefox 58.0
- Billing Care is Now Certified with Oracle Identity and Access Management 12c Release 2
- SDK for Customizing the Purchase Flow and Assets Display
- Support for Backdated Account Creation and Product Purchase
- Support for Creating Wholesale Accounts and Billing Hierarchies
- Support for Customizing Product Information Post Account Creation
- Support for Deferred Actions
- Support for Multischema
Support for Undefined Payment Method
Support for Enabling SSO using SAML
For more information on these enhancements, see the Billing Care Online Help.

Additional Options for Searching Bills

Billing Care now allows you to search for:
- Specific bills by using bill number
- The last generated bill
- Bills generated between specific dates by using date range

Billing Care Is Now Certified with Mozilla Firefox 58.0

Currently, Billing Care 12.0 is certified with Mozilla Firefox 54.0.1. With this patch, Billing Care 12.0 is certified with Mozilla Firefox 58.0.
For more information, see Billing Care Installation Guide.

Billing Care is Now Certified with Oracle Identity and Access Management 12c Release 2

Currently, Billing Care 12.0 is certified with Oracle Identity and Access Management (OIAM) 11g Release 2 (11.1.2.3.0). With this patch, Billing Care 12.0 is certified with OIAM 12c R2 (12.2.1.3.0).
For more information, see Billing Care Installation Guide.

SDK for Customizing the Purchase Flow and Assets Display

You can now customize the Purchase Catalogue screen, add custom fields, and customize the Assets area to display the custom fields by using the Billing Care SDK. For example, this SDK allows you hide notes, fields, and the purchase start date or end date in the Purchase Catalogue screen and view the newly added custom fields in the Assets area.
For more information on customizing Billing Care to customize Purchase Catalogue and Assets display, see Billing Care SDK Guide.

Support for Backdated Account Creation and Product Purchase

Billing Care now allows you to backdate the account creation and product purchase. You can now enter or select an account creation date or a product purchase date that is earlier than the current date.

Support for Creating Wholesale Accounts and Billing Hierarchies

Billing Care now allows you to create wholesale accounts and billing hierarchies. You can create wholesale parent and child accounts and set up a wholesale billing hierarchy by assigning the bill unit of the wholesale parent account to the wholesale
business profile and the bill unit of each wholesale child account to the wholesale parent account for payment.

Support for Customizing Product Information Post Account Creation

Billing Care now allows you to customize products after account creation by using the Rate Customizations option. For example, you can modify a package or bundle and the start date and end date for charges (such as recurring, one time, and usage charges).

Support for Deferred Actions

Billing Care now allows you to perform actions on the account, service, and collections-related deferred actions that are already created. By using the Deferred Actions option, you can edit, execute, and delete the deferred actions.

Support for Multischema

Billing Care now supports multischema. You can now view the accounts in the different schemas in Billing Care.

Support for Undefined Payment Method

Billing Care now allows you to create account and access services without defining any specific payment method. You can now set the payment method for a bill unit to No Payment Method when there is no payment method currently defined for an account or service. This enables you offer free trial for customers. You can define a specific payment method after the free period has ended.

Support for Enabling SSO using SAML

You can now use SAML to enable single-sign on (SSO) in Billing Care. SSO allows you to log in to applications using a single user name and password combination.

For more information, see the discussion about configuring SAML for SSO in Billing Care Installation Guide.
New Features in Business Operations Center

This chapter provides an overview of the new features in Oracle Communications Business Operations Center 12.0 Patch Set 1 and Patch Set 2.

New Features in Business Operations Center 12.0 Patch Set 2

Business Operations Center 12.0 Patch Set 2 includes the following enhancements:

- Workflow Job to Automate Billing Process

Workflow Job to Automate Billing Process

In previous releases, you had to create separate business operation center jobs for billing and invoicing and manage the dependency between the jobs using scheduled time.

In this case, if the jobs were not executed in the correct order, the bills could reflect incorrect charges.

With this enhancement, you can run a workflow job, which can include billing, payment collections, invoices, and refunds. Workflow jobs enable you to automate an end-to-end billing process.

Note:

Only BOC Super Admin can run workflow jobs.

For more information on the Admin role, see the discussion about roles in the Business Operations Center Help.

You can include one or more of these jobs in a workflow job, in this order:

- Billing
- Invoicing
- Payment collections, including the option to settle any previously authorized one-time payments.
- Refunds
Note:
You can't change the order. A workflow job stops if any individual job fails.

And, you can specify these attributes to determine which accounts are included:

- The status of the accounts to bill: active, inactive, or closed.
- The payment method.
  - Payment card (credit card or debit card)
  - Invoice
- The account’s billing day of month. You can run billing for accounts whose billing day of month is:
  - On or before the job is run. The job creates bills for accounts whose billing date is any day before midnight of the day that you run the job. If you run billing every day, the accounts billed are from the previous day. If you run billing every two weeks, all accounts with a billing date in the previous two weeks are billed.
  - A specified number of days before the job is run.
  - On a specific day of the month. You can enter multiple days.

If invoice jobs are included, specify the level of information to include in the invoices: summary or detailed.

You can view the workflow job status and metrics from the Job History page by clicking the link displayed under Processed in each row. However, you cannot rerun the failed workflow jobs from this page.

New Features in Business Operations Center 12.0 Patch Set 1

Business Operations Center 12.0 Patch Set 1 includes the following enhancements:

- Business Operations Center is Now Certified with Jersey Bundle 2.27 (Patch Set 1)
- Business Operations Center is Now Certified with Jackson 2.9.6 (Patch Set 1)
- Business Operations Center is Now Certified with Log4j2 2.11.1 (Patch Set 1)
- Business Operations Center Is Now Certified with Mozilla Firefox 58.0 (Patch Set 1)
- Business Operations Center is Now Certified with Oracle Identity and Access Management 12c Release 2 (Patch Set 1)
- Business Operations Center is Now Certified with Quartz Scheduler 2.3.0 (Patch Set 1)
Business Operations Center is Now Certified with Jersey Bundle 2.27 (Patch Set 1)

Currently, Business Operations Center 12.0 is certified with Jersey Bundle 2.25.1. With this patch, Business Operations Center 12.0 is certified with Jersey Bundle 2.27.

For more information, see Business Operations Center Installation Guide.

Business Operations Center is Now Certified with Jackson 2.9.6 (Patch Set 1)

Currently, Business Operations Center 12.0 is certified with Jackson version 2.8.9. With this patch, Business Operations Center 12.0 is certified with Jackson version 2.9.6.

For more information, see Business Operations Center Installation Guide.

Business Operations Center is Now Certified with Log4j2 2.11.1 (Patch Set 1)

Currently, Business Operations Center 12.0 is certified with Log4j2 2.8.2. With this patch, Business Operations Center 12.0 is certified with Log4j2 2.11.1.

For more information, see Business Operations Center Installation Guide.

Business Operations Center Is Now Certified with Mozilla Firefox 58.0 (Patch Set 1)

Currently, Business Operations Center 12.0 is certified with Mozilla Firefox 54.0.1. With this patch, Business Operations Center 12.0 is certified with Mozilla Firefox 58.0.

For more information, see Business Operations Center Installation Guide.

Business Operations Center is Now Certified with Oracle Identity and Access Management 12c Release 2 (Patch Set 1)

Currently, Business Operations Center 12.0 is certified with Oracle Identity and Access Management (OIAM) 11g Release 2 (11.1.2.3.0). With this patch, Business Operations Center 12.0 is certified with OIAM 12c R2 (12.2.1.3.0).

For more information, see Business Operations Center Installation Guide.

Business Operations Center is Now Certified with Quartz Scheduler 2.3.0 (Patch Set 1)

Currently, Business Operations Center 12.0 is certified with Quartz Scheduler 2.2.3. With this patch, Business Operations Center 12.0 is certified with Quartz Scheduler 2.3.0.
For more information, see *Business Operations Center Installation Guide*. 
7

Opcode Changes

This chapter provides an overview of the opcode changes introduced in Oracle Communications Billing and Revenue Management (BRM) 12.0 Patch Set 1 and Patch Set 2.

New Opcodes

This section describes the standard and policy opcodes introduced in BRM 12.0 patch sets.

New Policy Opcodes

This section describes the policy opcodes introduced in BRM 12.0 patch sets.

Subscription Management FM Policy Opcodes

Table 7-1 lists the new Balance framework standard opcodes.

Table 7-1 New Subscription Management FM Policy Opcodes

<table>
<thead>
<tr>
<th>New Policy Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM_OP_SUBSCRIPTION_POL_PRE_CYCLE_DISCOUNT</td>
<td>(Patch Set 1) Determines the balance elements (resources) of the nonpaying child bill units for rolling them up to the paying parent bill unit during wholesale billing. This opcode is called before applying the cycle discounts.</td>
</tr>
</tbody>
</table>

Changed Opcodes

This section describes the standard and policy opcodes changed in BRM 12.0 patch sets.

Changed Standard Opcodes

Table 7-2 lists the changed standard opcodes.

Table 7-2 Changed Standard Opcodes

<table>
<thead>
<tr>
<th>Changed Standard Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM_OP_AR_GET_BILLS</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_ACCT_BILLS</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_ACCT_BAL_SUMMARY</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_ACTION_ITEMS</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>Changed Standard Opcode</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_ACCT_ACTION_ITE_MS</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_BAL_SUMMARY</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_BILL_ITEMS</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_DISPUTES</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_AR_GET_DISPUTEDETAILS</td>
<td>(Patch Set 2) Modified to support enhanced wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_CUST_COMMIT_CUSTOMER</td>
<td>(Patch Set 2) Modified to include the PIN_FLD_TRANSACTIONS array for storing card credentials.</td>
</tr>
<tr>
<td>PCM_OP_CUST_CREATE_PAYINFO</td>
<td>(Patch Set 2) Modified to include the PIN_FLD_TRANSACTIONS array for storing card credentials.</td>
</tr>
<tr>
<td>PCM_OP_CUST_SET_PAYINFO</td>
<td>(Patch Set 2) Modified to include the PIN_FLD_TRANSACTIONS array for storing card credentials.</td>
</tr>
<tr>
<td>PCM_OP_PYMT_CHARGE</td>
<td>(Patch Set 2) Modified to use the stored card credentials for payment transactions.</td>
</tr>
<tr>
<td>PCM_OP_PYMT_CHARGE_CC</td>
<td>(Patch Set 2) Modified to use the stored card credentials for payment transactions.</td>
</tr>
<tr>
<td>PCM_OP_PYMT_COLLECT</td>
<td>(Patch Set 2) Modified to use the stored card credentials for payment transactions.</td>
</tr>
<tr>
<td>PCM_OP_AR_ACCOUNT_ADJUSTMENT</td>
<td>(Patch Set 1) Modified to skip account-level adjustment for the nonpaying child bill units in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td>PCM_OP_AR_ACCOUNT_WRITEOFF</td>
<td>(Patch Set 1) Modified to skip write-off adjustments for the nonpaying child bill units in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td>PCM_OP_AR_BILL_ADJUSTMENT</td>
<td>(Patch Set 1) Modified to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• Limit the search to only the paying parent bill unit in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td></td>
<td>• Perform the validations to ensure that the bill amount is not over-adjusted during the adjustment.</td>
</tr>
<tr>
<td>PCM_OP_AR_BILL_DISPUTE</td>
<td>(Patch Set 1) Modified to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• Limit the search to only the paying parent bill unit in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td></td>
<td>• Perform the validations to ensure that the dispute items of the nonpaying child bill units are not provided in the input flist.</td>
</tr>
<tr>
<td>PCM_OP_AR_BILL_SETTLEMENT</td>
<td>(Patch Set 1) Modified to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• Limit the search to only the paying parent bill unit in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td></td>
<td>• Perform the validations to ensure that the settlement items of the nonpaying child bill units are not provided in the input flist.</td>
</tr>
<tr>
<td>PCM_OP_AR_BILL_WRITEOFF</td>
<td>(Patch Set 1) Modified to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• Limit the search to only the paying parent bill unit in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td></td>
<td>• Skip write-off adjustments for the nonpaying child bill units in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td>PCM_OP_AR_BILLINFO_WRITEOFF</td>
<td>(Patch Set 1) Modified to limit the search to only the paying parent bill unit in the wholesale bill unit hierarchy and skip write-off adjustments for the nonpaying child bill units in the wholesale bill unit hierarchy.</td>
</tr>
</tbody>
</table>
### Table 7-2  (Cont.) Changed Standard Opcodes

<table>
<thead>
<tr>
<th>Changed Standard Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM_OP_AR_EVENT_ADJUSTMENT</td>
<td>(Patch Set 1) Modified to adjust a billed event's balance impact of a nonpaying child bill unit in the wholesale bill unit hierarchy. For wholesale billing, this opcode creates temporary A/R items (I \tmp_ar_item_to_roll_up object) with the details of the adjustment and the due amount of the nonpaying child bill unit after adjustment.</td>
</tr>
<tr>
<td>PCM_OP_AR_EVENT_DISPUTE</td>
<td>(Patch Set 1) Modified to open disputes for the billed events of the nonpaying child bill units in the wholesale bill unit hierarchy. For wholesale billing, this opcode creates temporary A/R items (I \tmp_ar_item_to_roll_up object) with the details of the dispute and the amount to be transferred to the paying parent bill unit.</td>
</tr>
<tr>
<td>PCM_OP_AR_EVENT_SETTLEMENT</td>
<td>(Patch Set 1) Modified to settle the billed dispute events of the nonpaying child bill units in the wholesale bill unit hierarchy. For wholesale billing, this opcode creates temporary A/R items (I \tmp_ar_item_to_roll_up object) with the details of the settlement and the latest disputed, transferred, and total amounts of the nonpaying child bill unit.</td>
</tr>
<tr>
<td>PCM_OP_AR_ITEM_ADJUSTMENT</td>
<td>(Patch Set 1) Modified to skip item-level adjustments for the billed items of the nonpaying child bill units in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td>PCM_OP_AR_ITEM_DISPUTE</td>
<td>(Patch Set 1) Modified to skip item-level dispute for the billed items of the nonpaying child bill units in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td>PCM_OP_AR_ITEM_SETTLEMENT</td>
<td>(Patch Set 1) Modified to settle the billed dispute items of the nonpaying child bill units in the wholesale bill unit hierarchy. For wholesale billing, this opcode creates temporary A/R items (I \tmp_ar_item_to_roll_up object) with the details of the settlement and the latest disputed, transferred, and total amounts of the nonpaying child bill unit.</td>
</tr>
<tr>
<td>PCM_OP_AR_ITEM_WRITEOFF</td>
<td>(Patch Set 1) Modified to skip write-off adjustments for the bill items of the nonpaying child bill units in the wholesale bill unit hierarchy.</td>
</tr>
<tr>
<td>PCM_OP_BILL_MAKE_BILL</td>
<td>(Patch Set 1) Modified to perform the following:</td>
</tr>
<tr>
<td></td>
<td>• roll charges (bill items) and journals of the nonpaying child bill units up to the paying parent bill unit for wholesale billing.</td>
</tr>
<tr>
<td></td>
<td>• Update the bill objects of the nonpaying child bill units for wholesale billing.</td>
</tr>
<tr>
<td>PCM_OP_COLLECTIONS_INVOKE_PROMISE_TO_PAY</td>
<td>(Patch Set 1) Modified to return the payment milestones, milestone amount, and milestone due date when called in the calc-only mode.</td>
</tr>
<tr>
<td>PCM_OP_CUST_COMMIT_CUSTOMER</td>
<td>(Patch Set 1) Modified to support wholesale account creation.</td>
</tr>
<tr>
<td>PCM_OP_CUST_CREATE_CUSTOMER</td>
<td>(Patch Set 1) Modified to support wholesale account creation.</td>
</tr>
<tr>
<td>PCM_OP_CUST_DELETE_ACCT</td>
<td>(Patch Set 1) Modified to delete all the objects (such as \event and \bill) and audit entries when the corresponding account is deleted.</td>
</tr>
</tbody>
</table>
Table 7-2  (Cont.) Changed Standard Opcodes

<table>
<thead>
<tr>
<th>Changed Standard Opcode</th>
<th>Description</th>
</tr>
</thead>
</table>
| PCM_OP_CUST_SET_BILLINFO | (Patch Set 1) Modified to perform the following:  
  • Set the items of the bill unit moved in or out of a wholesale bill unit hierarchy to point to the appropriate AR_ITEM_OBJ and ITEM_CLASS.  
  • Set the PIN_FLD_PARENT_FLAGS field to 1 only if the bill unit is assigned to the wholesale business profile or if the system-wide wholesale billing is enabled.  
  • Perform the validations to ensure that there is only one paying parent bill unit in the wholesale bill unit hierarchy. |
| PCM_OP_INV_MAKE_INVOICE | (Patch Set 1) Modified to create invoice only for the items of the paying parent bill unit in the wholesale bill unit hierarchy. |
| PCM_OP_INV_POL_PREP_INVOICE | (Patch Set 1) Modified to retrieve only the items of the paying parent bill unit in the wholesale bill unit hierarchy. |
| PCM_OP_PYMT_COLLECT | (Patch Set 1) Modified to consider only the items of the paying parent bill unit in the wholesale bill unit hierarchy. |
| PCM_OP_PYMT_MBI_DISTRIBUTE | (Patch Set 1) Modified to consider only the items of the paying parent bill unit in the wholesale bill unit hierarchy. |
| PCM_OP_SUBSCRIPTION_RERATE_REBILL | (Patch Set 1) Modified to create temporary A/R items (\texttt{tmp_ar_item_to_roll_up} object) after rerating nonpaying child bill units in the wholesale bill unit hierarchy. |

Changed Policy Opcodes

This section describes the policy opcodes changed in BRM 12.0 patch sets.

Changed Policy Opcodes

Table 7-3 lists the changed policy opcodes.

Table 7-3  Changed Policy Opcodes

<table>
<thead>
<tr>
<th>New Policy Opcode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCM_OP_PYMT_POL_PRE_COLLECT</td>
<td>(Patch Set 2) Modified to support PIN_FLD_TRANSACTIONS array for stored-credential transactions.</td>
</tr>
<tr>
<td>PCM_OP_CUST_POL_VALID_PAYINFO</td>
<td>(Patch Set 2) Modified to support PIN_FLD_TRANSACTIONS array for stored-credential transactions.</td>
</tr>
<tr>
<td>PCM_OP_INV_POL_PREP_INVOICE</td>
<td>(Patch Set 1) Modified to prepare the invoice information for wholesale accounts.</td>
</tr>
</tbody>
</table>
This chapter provides an overview of the storable class changes introduced in Oracle Communications Billing and Revenue Management (BRM) 12.0 Patch Set 1 and Patch Set 2.

### New Storable Classes

Table 8-1 lists the storable classes introduced in the BRM 12.0 patch sets. The new storable classes are grouped by the respective patch sets.

<table>
<thead>
<tr>
<th>New Storable Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/history_subscriber_contract</td>
<td>(Patch Set 2) Stores a copy of the subscriber contract data. This class is created when a contract is renewed or modified.</td>
</tr>
<tr>
<td>/subscriber_contract</td>
<td>(Patch Set 2) Stores the subscriber contract data.</td>
</tr>
<tr>
<td>/cmt_balances</td>
<td>(Patch Set 2) Stores the balance data temporarily.</td>
</tr>
<tr>
<td>/tmp_ar_item_to_roll_up</td>
<td>(Patch Set 1) Stores information for the A/R items of the nonpaying child bill units to roll them up to the paying parent bill unit.</td>
</tr>
<tr>
<td>/event/activity/roll_up</td>
<td>(Patch Set 1) Records activity related to rolling of items up to the paying parent bill unit.</td>
</tr>
<tr>
<td>/recurring_bundle_history</td>
<td>(Patch Set 1) Stores information about recurring bundles.</td>
</tr>
</tbody>
</table>

### Changed Storable Classes

Table 8-1 lists the storable classes changed in the BRM 12.0 patch sets. The changed storable classes are grouped by the respective patch sets.

<table>
<thead>
<tr>
<th>Changed Storable Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/event/billing/charge/cc</td>
<td>(Patch Set 2) Modified to support stored-credential transactions.</td>
</tr>
<tr>
<td>/event/billing/validate/cc</td>
<td>(Patch Set 2) Modified to support stored-credential transactions.</td>
</tr>
<tr>
<td>/event/customer/nameinfo</td>
<td>(Patch Set 2) Modified to support stored-credential transactions.</td>
</tr>
<tr>
<td>/job/boc</td>
<td>(Patch Set 2) Modified to run workflow jobs in Business Operations Center.</td>
</tr>
<tr>
<td>Changed Storable Class</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/job_template/billing</td>
<td>(Patch Set 2) Modified to run workflow jobs in Business Operations Center.</td>
</tr>
<tr>
<td>/job_template/collect</td>
<td>(Patch Set 2) Modified to support workflow jobs in Business Operations Center.</td>
</tr>
<tr>
<td>/job_template/invoice</td>
<td>(Patch Set 2) Modified to support workflow jobs in Business Operations Center.</td>
</tr>
<tr>
<td>/job_template/ledger_report</td>
<td>(Patch Set 2) Modified to support workflow jobs in Business Operations Center.</td>
</tr>
<tr>
<td>/payinfo/cc</td>
<td>(Patch Set 2) Modified to store card credentials for future transactions.</td>
</tr>
<tr>
<td>/rate</td>
<td>(Patch Set 2) Modified to support enhancements in BRM 12.0 Patch Set 2.</td>
</tr>
<tr>
<td>/service</td>
<td>(Patch Set 2) Modified to support enhancements in BRM 12.0 Patch Set 2.</td>
</tr>
<tr>
<td>/tmp_journals_to_process</td>
<td>(Patch Set 1) Modified to store information for the journals of the nonpaying child bill units to roll them up to the paying parent bill unit.</td>
</tr>
</tbody>
</table>
Schema and Index Changes

This chapter provides an overview of the schema and index changes introduced in Oracle Communications Billing and Revenue Management (BRM) 12.0 Patch Set 1 and Patch Set 2.

BRM Schema and Index Changes

This section describes the schema and index changes in BRM.

Schema and Index Changes (Patch Set 2)

BRM 12.0 Patch Set 2 includes the following schema and index changes:

- The following tables are added:
  - CMT_BALANCES_T
  - EVT_BILLING_CHARGE_CC_TRANS_T
  - EVT_BILL_VLDT_CC_TRANS_T
  - JOB_TEMPLATE_TBILL_BILLINFO_T
  - JOB_TEMPLATE_TBILL_INFO_T
  - JOB_TEMPLATE_TBILL_PAY_TYPES_T
  - JOB_TEMPLATE_TBILL_STATUSES_T
  - JOB_TEMPLATE_TBILLP_BILLINFO_T
  - JOB_TEMPLATE_TBILLP_INFO_T
  - JOB_TEMPLATE_TBILLP_STATUSES_T
  - JOB_TEMPLATE_WF_CATEGORIES_T
  - NAMEINFO_T
  - PAYINFO_CC_TRANS_T

- The following columns are added in the EVENT_CUSTOMER_NAMEINFO_T table:
  - POID_DB
  - POID_ID0
  - POID_TYPE
  - POID_REV

- The following columns are added in the JOB_BOC_T table:
  - WORKFLOW_OBJ_ID
  - WORKFLOW_OBJ_TYPE
  - WORKFLOW_OBJ_REV
• The following column is added in the JOB_TEMPLATE_T table:
  – WORKFLOW_FLAG

• The following columns are added in the HISTORY_SUBSCRIBER_CONTRACT_T table:
  – GROUP_CONTRACT_OBJ_ID0
  – GROUP_CONTRACT_OBJ_TYPE
  – GROUP_CONTRACT_OBJ_REV
  – PLAN_CONTRACT_OBJ_DB
  – PLAN_CONTRACT_OBJ_ID0
  – PLAN_CONTRACT_OBJ_TYPE
  – PLAN_CONTRACT_OBJ_REV

• The following column is added in the RATE_BAL_IMPACTS_T table:
  – TAXABLE_AMOUNT

• The following columns are added in the SERVICE_T table:
  – NAMEINFO_OBJ_ID0
  – NAMEINFO_OBJ_TYPE
  – NAMEINFO_OBJ_REV

• The following columns are added in the SUBSCRIBER_CONTRACT_T table:
  – GROUP_CONTRACT_OBJ_DB
  – GROUP_CONTRACT_OBJ_ID0
  – GROUP_CONTRACT_OBJ_TYPE
  – GROUP_CONTRACT_OBJ_REV
  – PLAN_CONTRACT_OBJ_DB
  – PLAN_CONTRACT_OBJ_ID0
  – PLAN_CONTRACT_OBJ_TYPE
  – PLAN_CONTRACT_OBJ_REV

• The following indexes are added:
  – I_PROC_AUD_POCCESS_END__ID
  – I_NAMEINFO__ID
  – I_NAMEINFO_ACCOUNT_OBJ__ID
  – I_NAMEINFO_PHONES__ID
  – I_PROFILE_SUBS_PREF__ID
  – I_ALLOWANCE__ID
  – I_ASS_BUS_PROFILE_TEMPLATES__ID
  – I_CFG_PROV_TAX_CODE_EFFECT__ID
  – I_CFG_PROV_TAXES_REC__ID
  – I_CFG_PROV_TAX_INFO__ID
• The following tables are added:
The following columns are added in the ITEM_T table:

- ITEM_CLASS
- AR_ITEM_OBJ

The following column is added in the TMP_JOURNALS_TO_PROCESS_T table:

- AR_BILLINFO_OBJ

The following indexes are added:

- I_TMP_AR_ITM_ROLLUP__ID
- I_TMP_AR_ITM_ROLLUP__STATUS
- I_ITEM_AR_ITEM_OBJ__ID
- I_TMP_JOURNALS_TO_PROCESS__AR
10 Utility Changes

This chapter provides an overview of the utility changes introduced in Oracle Communications Billing and Revenue Management (BRM), Business Operations Center, Pipeline Configuration Center (PCC), Oracle Communications Billing Care, and Oracle Communications Pricing Design Center (PDC) 12.0 Patch Set 1 and Patch Set 2.

BRM Utilities

This section lists the utilities added or changed in BRM.

New Utilities

Table 10-1 lists the utilities introduced in BRM 12.0 patch sets.

<table>
<thead>
<tr>
<th>New Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pin_del_closed_accts</td>
<td>(Patch Set 1) Introduced to delete closed accounts in BRM. This utility calls the PCM_OP_CUST_DELETE_ACCT opcode to delete the closed accounts that are older than the specified retention period. See “Enhanced Data Protection” for more information.</td>
</tr>
<tr>
<td>pin_roll_up_ar_items</td>
<td>(Patch Set 1) Introduced to process the temporary A/R items (I tmp_ar_item_to_roll_up object) of the nonpaying child bill units and roll the balance impact up to the corresponding A/R items in the paying parent bill unit. You can run multiple threads of pin_roll_up_ar_items to process A/R items for different paying parent bill unit. See “Improved Performance for Large Accounts” for more information.</td>
</tr>
<tr>
<td>pin_update_journal</td>
<td>(Patch Set 1) Enhanced to process the temporary journals (I tmp_journals_to_process object) of the nonpaying child bill units and roll them up to the paying parent bill unit. You must run this utility before billing the paying parent bill unit. See “Improved Performance for Large Accounts” for more information.</td>
</tr>
</tbody>
</table>

Changed Utilities

Table 10-2 lists the utilities changed in BRM 12.0 patch sets.
### Table 10-2  Changed Utilities

<table>
<thead>
<tr>
<th>New Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmt_mta_cycle_fees</td>
<td>(Patch Set 2) Renamed as <strong>cmt_mta_deploy</strong> and modified to support incremental migration of legacy BRM data. See “Support for Migration of Legacy Data into BRM and ECE in Real Time” for more information.</td>
</tr>
<tr>
<td>partition_utils</td>
<td>(Patch Set 2) Modified to support creation of partitions for the prepaid events. See “BRM Supports POID Generation in ECE” for more information.</td>
</tr>
<tr>
<td>pin_cmt</td>
<td>(Patch Set 2) Modified to migrate legacy service and balance data incrementally into the BRM system. You can now run the <strong>pin_cmt</strong> utility with the following new parameters to migrate data:</td>
</tr>
<tr>
<td></td>
<td>• <strong>deploy_db</strong></td>
</tr>
<tr>
<td></td>
<td>• <strong>deploy_ece</strong></td>
</tr>
<tr>
<td></td>
<td>See “Support for Migration of Legacy Data into BRM and ECE in Real Time” for more information.</td>
</tr>
<tr>
<td>pin_collect</td>
<td>(Patch Set 2) Modified to support stored credentials for payment transactions. See “Support for Stored-Credential Transactions for Payments” for more information.</td>
</tr>
<tr>
<td>pin_deposit</td>
<td>(Patch Set 2) Modified to support stored credentials for payment transactions. See “Support for Stored-Credential Transactions for Payments” for more information.</td>
</tr>
</tbody>
</table>

### Removed Utilities

Table 10-3 lists the utilities that were removed in BRM 12.0 patch sets.

### Table 10-3  Removed Utilities

<table>
<thead>
<tr>
<th>New Utility</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pin_update_items_journals</td>
<td>(Patch Set 1) This utility has been replaced by the <strong>pin_update_journal</strong> utility. See “Improved Performance for Large Accounts” for more information.</td>
</tr>
</tbody>
</table>
11

Known Problems

This chapter provides an overview of the known problems in Oracle Communications Billing and Revenue Management (BRM), Business Operations Center, Pipeline Configuration Center (PCC), Oracle Communications Billing Care, and Oracle Communications Pricing Design Center (PDC) 12.0 Patch Set 1 and Patch Set 2.

Known Problems in BRM

This section describes known problems and workarounds for BRM 12.0 patch sets.

A/R Actions between Billing and Invoicing Not Captured in Invoice

The invoicing process does not pick up the details of any A/R action that occurs between the time when a bill is generated and when the invoice for that bill is produced.

**Workaround:** Run billing and invoicing as an atomic operation without other A/R actions in between. For more information, see *BRM Configuring and Running Billing*.

Bill Now Generates Two Audit Objects for Nonpaying Child Items

When you run Bill Now on an account, selecting items corresponding to a nonpaying child bill unit, two audit objects are created: one audit object with an amount of 0 as revenue for the account that owns the paying parent bill unit and another audit object for the account that owns the nonpaying child bill unit.

Currently, there is no workaround.

Default Invoices Do Not Display Custom Events and Items

The default BI Publisher invoice templates shipped with BRM do not display data from custom /event and /item objects.

**Workaround:** If you create custom /event or /item objects, you must also customize the BI Publisher invoice templates to display your custom data. For more information, see the discussion about customizing invoices in *BRM Designing and Generating Invoices*.

Error When Configuring or Modifying Security Policies for Web Services Manager for JAX-WS

If you have configured security policies for Web Services Manager for JAX-WS or if you have modified the security policies, the request sent by the client to the server fails and the following error message appears:

headers:[http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd]Security] are not understood
Workaround: If your client caches service WSDLs, refresh the WSDLs before sending the requests to the server.

Event Browser Displays Data Internal to BRM in the Event Description

For usage events that were rated by Oracle Communications Elastic Charging Engine (ECE), Event Browser displays data internal to BRM in the event description.

Currently, there is no workaround.

Invoices Do Not Display Charge Sharing Details Separately

The balances associated with charge sharing are not split into original charges and sponsored/shared charges in the invoices. The total balances are correct and include any effect of charge sharing.

Currently, there is no workaround.

Invoices Might Display Wrong Discount Amount

Invoices might display the wrong discount amount for accounts with nonpaying bill units in a discount sharing group.

Currently, there is no workaround.

pin_inv_accts Fails for Migrated Accounts if Business Intelligence Publisher Invoicing Integration Is Enabled

If Oracle Business Intelligence Publisher (BI Publisher) invoicing integration is enabled, the `pin_inv_accts` utility fails for migrated accounts and an error is logged in the `cm.pinlog` file.

Workaround: Do the following before migrating the accounts:

1. Open the `BRM_home/sys/amt/data/create_generate_amt_metadata.sql` file in a text editor.
2. Search for the following lines:

   ```sql
   ELSIF (column_r.column_name LIKE '%POID_LIST') THEN
   --
   -- operation: rewrite the POID_DB reference in %POID_LIST fields
   -- (i.e. account_t.next_item_poid_list, account_t.item_poid_list)
   -- while preserving the null value
   --
   -- assumption: POID_DB reference is starts at character position 1 and the format is 0.0.0.x
   --
   -- call custom Java stored procedure convert_poid_list()
   --
   select_column_list := select_column_list||'
   ', convert_poid_list(''||column_r.column_name||'
   ', #);
   dbms_output.put_line('replace list reference with dest_db_id');
   ```

3. Add the following lines after the ELSIF condition you searched for in step 2:

   ```sql
   ELSIF (column_r.column_name LIKE '%OBJ_LIST') THEN
   select_column_list := select_column_list||
   ```
4. Save and close the file.
5. Run the `pin_amt_install.pl` script.
6. Stop and start the Data Manager (DM) and Connection Manager (CM) services.
7. Start migrating the accounts.

Refund Is Incorrectly Calculated When Remaining Charge Discount Canceled

On canceling a remaining charge discount on a cycle forward arrears event, the refund amount is calculated incorrectly. Currently, there is no workaround.

Sample Prepaid Service Life Cycle Does Not Support Provisioning of Telco Services

The sample prepaid service life cycle does not support provisioning of telco services.

**Workaround:** Enable the sample prepaid service life cycle to support the provisioning of telco services by adding the following state transition to that life cycle before associating the life cycle with a telco service type:

- From state: Preactive (ID 101)
- To state: Suspended (ID 107)

For information about adding state transitions to service life cycles, see the discussion about managing service life cycles in *BRM Managing Customers*.

The SLM Feature Does Not Support Account Billing Relationships

The SLM feature does not support the following account relationships, which enable customers to pay other customers' bills:

- Parent/nonpaying child hierarchies
- Owner/member charge sharing or discount sharing
- Owner/member sponsorship

If a service involved in such a relationship uses a custom life cycle, the SLM feature does not consider the condition of the parent/owner account when an event occurs that triggers a service state change in a child/member account.

For example, Service A uses the sample prepaid service life cycle. For that service, a parent account has a credit limit of 100 euros and its nonpaying child account has a credit limit of 0 euros. In both the parent and the child, the state of Service A is Active. After an account adjustment of 2 euros is applied to the child account, however, the state of Service A in the child account changes to Recharge Only, even though the parent account's available credit limit is still 100 euros.
Therefore, service types involved in account billing relationships must use the default service life cycle (Active, Inactive, Closed). They cannot use custom service life cycles.

**Workaround:** Customize the PCM_OP_BAL_POL_CHECK_LIFECYCLE_STATE policy opcode to consider the condition of parent/owner accounts when a service state change is triggered in child/member accounts.

See the discussion about managing service life cycles in *BRM Managing Customers* for more information.

**Settlement Taxes Might Be Incorrectly Calculated**

If you adjust only the tax amount in an event (by calling the PCM_OP_AR_EVENT_ADJUSTMENT opcode with the PIN_EVENT_ADJ_TAX_ONLY flag) and then dispute the item containing the event, the settlement tax is not calculated correctly; therefore the total due amount of the item is incorrect. The settlement is calculated correctly only when the denied amount is 0. This occurs with both real-time and deferred taxation.

Currently, there is no workaround.

**stop_eai_js Does Not Work**

When you run `stop_eai_js`, the script does not work and the following error message appears:

`stop_eai_js: ERROR: no pid file`.

**Workaround:** Do the following:

1. Open the `BRM_home/bin/stop_eai_js` file in a text editor.
2. Search for the following entry:
   
   \[ \text{LOGDIR}=$\text{PIN_LOG/dm_eai} \]

3. Change this entry to:
   
   \[ \text{LOGDIR}=$\text{PIN_LOG/eai_js} \]

4. Save and close the file.

**Summary Invoices Do Not Show Real-Time Discount and Tax Details for Items**

Summary invoices display data related to charges at the item level. Details of real-time taxes or real-time discounts are not displayed separately on summary invoices. The charges for the billable items include this data.

**Workaround:** To view these details on invoices, generate the invoices in one of the following ways:

- Use the detailed format.
- Customize the item assignment configuration to aggregate the discounts and taxes into their own separate items.
The invoices will contain separate lines for the discounts and taxes. For more information, see *BRM Designing and Generating Invoices*.

**Tax Not Recalculated after Rerating**

When you rerate the bill amount after rerating the changes, the tax amount is not recalculated.

**Workaround**: manually adjust the tax amount.

**Known Problems in Business Operations Center**

There are no known problems in Business Operations Center 12.0 patch sets.

**Known Problems in Billing Care**

This section describes known problems and workarounds for Business Operations Center 12.0 patch sets.

**Duplicate Login Request When Using Single Sign-On**

When Single Sign-On (SSO) with Oracle Identity and Access Management Lifecycle Management is implemented for your Billing Care environment, users are redirected to the Billing Care login screen after authenticating in the SSO login screen.

**Workaround**: On each Oracle Platform Security Services (OPSS) JRF-enabled domain where Billing Care is deployed:

1. Comment out the `<variable-definition>` and `<module-override>` entries in the `Middleware_home/setup/Plan.xml` file as shown below, where `Middleware_home` is the WebLogic home directory on the OES Client domain host of the user who installed Billing Care.

```xml
<application-name>BillingCare.war</application-name>
<variable>
  <name>CONFIG_SSL_REDIRECT</name>
  <value>CONFIDENTIAL</value>
</variable>
</variable-definition>
</deployment-plan>
```

<?xml version='1.0' encoding='UTF-8'?>
<deployment-plan xmlns="http://xmlns.oracle.com/weblogic/deployment-plan"
<application-name>BillingCare.war</application-name>

<!--<variable-definition>
  <variable>
    <name>CONFIG_SSL_REDIRECT</name>
    <value>CONFIDENTIAL</value>
  </variable>
</variable-definition>-->
2. Restart the domain.

Firefox and Internet Explorer Fail to Connect Securely to Billing Care

The Plan.xml file deployed by the Billing Care Oracle Universal Installer enables mandatory use of an SSL-enabled port when connecting with all browsers. Firefox and Internet Explorer may fail to connect to Billing Care, producing a secure connection failure error.

Workaround: For Firefox, no workaround exists. For Internet Explorer, start a Command Prompt session with administrative rights, and run the following command before launching Internet Explorer:

certutil -setreg chain\EnableWeakSignatureFlags 8

Known Problems in PDC

This section describes known problems and workarounds for PDC 12.0 patch sets.

PDC Search Functionality Not Working As Expected

Basic Search

When you perform a basic search in the PDC user interface, clicking a search result does not open it.

Workaround: Double-click a search result to open it.

Advanced Search

If you are using Internet Explorer 11 on Windows 10, the search results do not appear as expected when you perform an Advanced search in the PDC user interface.

Workaround: Use Mozilla Firefox on Windows 10 or use Internet Explorer 11 on Windows 8.1 to perform Advanced search in the PDC user interface.

Migrating Pricing Data Fails in a PDC System with BRE and RRE

When the zone result validity period set in a zone model exceeds the validity period set for the associated rate plan, migration fails with an validation error.

Workaround: Modify the zone result validity in BRM to ensure that the zone result validity overlaps or falls within the associated rate plan validity.
Adding Balance Groups and Setting Credit Limits in a Package
Displays Exceptions

**Setting Credit Limits Exception**
In the Balance Groups section of the Create Package page, when you set the credit limit for a balance element and save it, the changes are saved successfully but PDC displays an exception.

**Workaround:** Navigate to any other section in the Create Package page and save the changes to avoid exceptions.

**Adding Balance Groups Exception**
When you add another balance group immediately after setting the credit limit for a balance element, the balance group is created successfully but PDC displays an exception instead of displaying the newly created balance group.

**Workaround:** Click the existing balance group to view the newly created balance group.

Migration Fails Even When the Migration Process is Enabled to Skip Errors

During migration, when a critical error is encountered while migrating a discount filter or a discount trigger, migration fails with errors even when the migration process is enabled to skip errors and continue migration.

**Workaround:** Manually fix the errors encountered during migration and restart the migration process.

You Are Not Warned to Save Your Changes

After modifying a setup or pricing component, you are not warned that you have unsaved changes when you do the following:

- Close the tab
- Log out of PDC

If you perform any of the above actions before saving your changes, the changes are lost.

**Workaround:** Save your changes before closing a tab or logging out of PDC.

**Known Problems in PCC**

There are no known problems in PCC 12.0 patch sets.