

**Oracle® Communications Billing and Revenue
Management**

Elastic Charging Engine 11.3 Release Notes

Release 7.5

E70773-09

November 2018

Oracle Communications Billing and Revenue Management Elastic Charging Engine 11.3 Release Notes, Release 7.5

E70773-09

Copyright © 2016, 2018, Oracle and/or its affiliates. All rights reserved.

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

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

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Accessing Oracle Communications Documentation	vii
Document Revision History	vii
1 Release Notes	
ECE 11.3 Patch Set 9 Release Notes	1-1
New Features in ECE 11.3 Patch Set 9	1-1
ECE Now Generates POID for Events	1-1
Automatic Retry of Failed Customer Updater Startup	1-4
ECE Now Monitors Diameter Peer Connections	1-5
ECE Now Supports Wildcard in Item Type Selectors	1-6
High Availability Support for ECE Components	1-6
Improved BRM-to-ECE Data Synchronization	1-9
Summary of Fixes in ECE 11.3 Patch Set 9	1-10
Known Problems in ECE 11.3 Patch Set 9	1-12
Pricing Updater Configured for High Availability Is Not Working As Expected	1-12
ECE 11.3 Patch Set 8 Release Notes	1-12
New Features in ECE 11.3 Patch Set 8	1-12
ECE Now Sends Group Notifications to Subscribers	1-13
Diameter Gateway Configuration Now Includes SCTP Attributes	1-14
Diameter Gateway Now Supports Both Incremental and Cumulative Accounting	1-16
Inactive and Dormant Accounts are Now Purged from the ECE Cache	1-17
Post-Rating Extension Can Now be Used to Add or Delete Rating Periods	1-18
Summary of Fixes in ECE 11.3 Patch Set 8	1-18
Known Problems in ECE 11.3 Patch Set 8	1-19
ECE 11.3 Patch Set 7 Release Notes	1-19
New Features in ECE 11.3 Patch Set 7	1-20
ECE Now Supports Incremental Rating for Tariff Changes	1-20
ECE Now Supports Dynamic Quota Allocation for Parallel Sessions	1-21
ECE Extensions Can be Used to Trigger RAR Notifications for Ongoing Sessions	1-22
ECE Now Monitors Diameter Statistics	1-23
BRM Gateway Now Automatically Processes the Failed Update Requests	1-25
ECE Now Uses Policy Specifications for Policy-Driven Charging	1-26

Summary of Fixes in ECE 11.3 Patch Set 7	1-26
Known Problems in ECE 11.3 Patch Set 7.....	1-29
ECE 11.3 Patch Set 6 Release Notes.....	1-29
New Features in ECE 11.3 Patch Set 6.....	1-29
Improved Rating Performance.....	1-29
Summary of Fixes in ECE 11.3 Patch Set 6	1-29
Known Problems in ECE 11.3 Patch Set 6.....	1-31
ECE 11.3 Patch Set 5 Release Notes.....	1-31
New Features in ECE 11.3 Patch Set 5.....	1-31
Support for Monitoring ECE	1-31
Pricing and Configuration Data Can Now Be Persisted in Active Persistence Mode ...	1-40
ECE Enhancement to Process Parallel Requests for the Same Customer	1-41
ECE Now Supports Match Factor in Discount Calculation.....	1-42
High Availability Support for ECE Driver Machine	1-42
Additional Attributes for Reservation Quota Management.....	1-43
Pre-Rating Extension Enhanced to Support Dynamic Quota.....	1-44
Post-Charging Extension Enhanced to Support URL Redirection	1-44
Service Transfer Data Updates Are Now Synchronized in Real Time	1-44
Support for GC Logs.....	1-44
Summary of Fixes in ECE 11.3 Patch Set 5	1-45
Known Problems in ECE 11.3 Patch Set 5.....	1-48
ECE 11.3 Patch Set 4 Release Notes.....	1-48
New Features in ECE 11.3 Patch Set 4.....	1-48
Diameter Gateway Enhancement for Policy Counters.....	1-48
ECE Now Supports Incremental Rating for Mid-session Rating Condition Changes ..	1-49
ECE Now Supports Subscriber-Based Tracing and Logging	1-51
Log4j Changes for Configuring Logging.....	1-57
Enhanced Extension Points to Enrich and Filter All External Notifications	1-58
Sessions Can Now Be Charged Based on the Session Connect Time.....	1-58
Summary of Fixes in ECE 11.3 Patch Set 4	1-59
Documentation Updates in ECE 11.3 Patch Set 4.....	1-61
Known Problems in ECE 11.3 Patch Set 4.....	1-62
ECE 11.3 Patch Set 3 Release Notes.....	1-62
New Features in ECE 11.3 Patch Set 3.....	1-62
Additional Charging Result Attributes Can Be Used for Item Assignment	1-62
Advice of Charge Notifications Now Include Quantity Details	1-62
Additional Data Storage Node Connections Can Be Configured for High Availability of Oracle NoSQL Database	1-63
CER Message Now Contains a List of Configured Vendor IDs.....	1-64
Configurable Log Location for Storing ECE Log Files	1-64
ECE Extension Point for Post-Update to Enrich and Filter Out External Notifications ..	1-64
ECE Now Supports Selective Loading of Data from BRM into ECE	1-65
KeepAlive Enabled for Connections Between BRM Connection Manager and EM Gateway 1-67	
New Reason Code Introduced to Indicate Rating Failure	1-67
Summary of Fixes in ECE 11.3 Patch Set 3	1-68
Known Problems in ECE 11.3 Patch Set 3.....	1-71
ECE 11.3 Patch Set 2 Release Notes.....	1-71

New Features in ECE 11.3 Patch Set 2.....	1-71
ECE Can Now Process Delayed Usage Requests for the Current Accounting Cycle....	1-71
ECE Can Now Determine Rates Based on Both the Calling and the Called Customers' Data	1-71
ECE Can Now Override Tax Rates or Assign Bill Items Based on the Extensions Data	1-72
ECE Can Now Use Shared Profiles to Determine a Price	1-72
Enhanced ECE Charging Functionality	1-72
Support for Using Rounding Rules Configured in PDC to Round the Charging Results.....	1-73
Support for Using Default System Currency for Charging.....	1-73
Summary of Fixes in ECE 11.3 Patch Set 2	1-73
Known Problems in ECE 11.3 Patch Set 2.....	1-75
ECE 11.3 Patch Set 1 Release Notes	1-75
New Features in ECE 11.3 Patch Set 1.....	1-75
ECE Credit Threshold and Credit Limit Breach Notifications Now Include Additional	
Information	1-75
ECE Now Supports Sending Rated Events to a Non-BRM System.....	1-76
ECE Now Supports the BRM Gateway Suspense Queue Configuration	1-76
ECE Now Supports Processing Offline Charging Requests for Accounts for Which Billing	
Has Been Delayed	1-76
Enhanced ECE External Notifications	1-77
Support for Changing ECE's Current Time and Date to Test ECE	1-77
Summary of Fixes in ECE 11.3 Patch Set 1	1-77
Known Problems in ECE 11.3 Patch Set 1.....	1-79
Rolling Upgrade Does Not Work for Upgrading to ECE 11.3 Patch Set 1	1-79
ECE 11.3 Release Notes	1-80
New Features in ECE 11.3.....	1-80
ECE Now Has RADIUS Gateway to Support RADIUS Protocols.....	1-80
ECE Now Supports Disaster Recovery	1-81
BRM-to-ECE Data Updates Are Now Synchronized in Real Time	1-81
ECE Can Now Generate Midsession Rated Events	1-82
ECE Supports Credit Control Requests Without Multiple-Service Credit Control	
Attribute-Value Pairs	1-82
ECE Now Supports Customer-Level Charge Offers, Alteration Offers, and Distribution	
Offers	1-82
ECE Now Supports Incremental Loading of Customer Data	1-82
Documentation Additions	1-83
Known Problems in ECE 11.3.....	1-83
SLM Feature Configuration Is Not Automatically Loaded into ECE	1-83

Preface

This guide includes information about this release of Oracle Communications Billing and Revenue Management (BRM) Elastic Charging Engine (ECE) 11.3 patch sets. Detailed information on the patch set features for ECE 11.3 Patch Set 3 through ECE 11.3 Patch Set 9 is included in this guide.

Detailed information on the patch set features for ECE 11.3 Patch Set 1 and ECE 11.3 Patch Set 2 is included in the ECE 11.3 documentation.

Audience

This guide is intended for all ECE users.

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.

Accessing Oracle Communications Documentation

ECE 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>

Document Revision History

The following table lists the revision history for this book.

Version	Date	Description
E70773-01	April 2016	Initial release.
E70773-02	September 2016	Added information about ECE 11.3 Patch Set 1 features.
E70773-03	December 2016	Added information about ECE 11.3 Patch Set 2 features.
E70773-04	April 2017	Added information about ECE 11.3 Patch Set 3 features.
E70773-05	August 2017	Added information about ECE 11.3 Patch Set 4 features.
E70773-06	December 2017	Added information about ECE 11.3 Patch Set 5 features.
E70773-07	March 2018	Added information about ECE 11.3 Patch Set 6 and ECE 11.3 Patch Set 7 features.
E70773-08	July 2018	Added information about ECE 11.3 Patch Set 8 features.
E70773-09	November 2018	Added information about ECE 11.3 Patch Set 9 features.

Release Notes

This chapter includes information about Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) 11.3 patch sets. Detailed information on the new features for ECE 11.3 Patch Set 3 through ECE 11.3 Patch Set 9 is included in this chapter.

Detailed information on the new features for ECE 11.3 Patch Set 1 and ECE 11.3 Patch Set 2 is included in the ECE 11.3 documentation.

ECE 11.3 Patch Set 9 Release Notes

This section provides information about ECE 11.3 Patch Set 9.

New Features in ECE 11.3 Patch Set 9

This section provides documentation for the features introduced in ECE 11.3 Patch Set 9.

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 1-1](#) for the description of each entry in the POID.

Table 1-1 POID Entries in ECE

Entry	Description
<i>event_type</i>	<p>A unique 4-bit number assigned to each event type.</p> <p>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 prepaidPartitionSet value specified in BRM.</p> <p>The default value for <i>event_type</i> is 0.</p>

Table 1–1 (Cont.) POID Entries in ECE

Entry	Description
<i>date</i>	The 16-bit date on which the POID is generated. The date is determined based on ECE virtualTime if it is enabled. For more information on virtualTime , see the discussion about changing time and date to test ECE in <i>BRM Elastic Charging Engine Implementation Guide</i> .
<i>cluster_id</i>	A unique 4-bit number assigned to the Coherence cluster to identify ECE in the cluster. The <i>cluster_id</i> is limited to 0 to 15 and the maximum number of ECE clusters allowed in a deployment is 16. The default value for <i>cluster_id</i> 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.
<i>BRM_schema_id</i>	A unique 6-bit number assigned to the BRM schema. The <i>BRM_schema_id</i> is limited to 0 to 31.
<i>unique_id</i>	A unique 34-bit number assigned to each POID.

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 the discussion about enabling prepaid-partition in *BRM 7.5 Patch Set 22 Release Notes*.
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/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.
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 **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.
2. Expand **charging.ratedEventFormatters.Instance_Name**, where *Instance_Name* is the name of the instance you want to configure; for example, `ratedEventFormatter2`.
3. Expand **Attributes**.
4. Set the **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 `ratedEventFormatter1`, specify `ratedEventFormatter1` as **primaryInstanceName** in the primary and all secondary instances.
5. Change directory to the *ECE_home/occeserver/bin* directory.
6. Start ECC:


```
./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 *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.
2. Expand **charging.brmCdrPlugins.Instance_Name**, where *Instance_Name* is the name of the BrmCdrPluginDirect Plug-in instance you are configuring.
3. Expand **Attributes**.
4. 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.

Automatic Retry of Failed Customer Updater Startup

In the previous releases, if Customer Updater failed to connect to the BRM database during startup, restart, or failover, you had to manually restart Customer Updater.

With this enhancement, you can configure Customer Updater to automatically retry the BRM database connection and restart the process by specifying the number of retries allowed after it fails. This ensures that Customer Updater is started automatically without any manual intervention.

To specify the retry count for Customer Updater:

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.
2. Expand **charging.connectionConfigurations.customerUpdater n** , where n is the number that represents the instance; for example, CustomerUpdater2.

3. Expand **Attributes**.

4. Set the **retryCount** attribute to the number of times the database connection can be retried after Customer Updater fails.

This attribute is applicable only for the Customer Updater startup, restart, or failover.

ECE Now Monitors Diameter Peer Connections

The ECE Monitoring Agent now monitors all the Diameter peers connected to the Diameter Gateway instances at regular intervals and logs the details of the peers (including the transport protocol) in the **monitorAgentX_ECE_SUMMARY_REPORT.log** file. By default, this log file is stored in the *ECE_home/occeserver/logs* directory. For more information on the Monitoring Agent, see ["Support for Monitoring ECE"](#).

The following is an example of the summary log generated for the Diameter peers:

```

"DiameterGatewayStatistics" : {
  ...
  "PeerConnections" : [
    { "DiameterGatewayName" : "diameterGateway1", "DiameterGatewayRealm" :
      "example.com", "DiameterGatewayIpAddress" : "127.0.0.1", "DiameterGatewayPort" :
      "3868", "PeerName" : "ccr.seagull.client.dgw1.peer2", "PeerRealm" : "example.com",
      "PeerIpAddress" : "127.0.0.1", "PeerPort" : "52247", "TransportProtocol" : "SCTP"
    }
    ,
    { "DiameterGatewayName" : "diameterGateway1", "DiameterGatewayRealm" :
      "example.com", "DiameterGatewayIpAddress" : "127.0.0.1", "DiameterGatewayPort" :
      "3868", "PeerName" : "ccr.seagull.client.dgw1.peer1", "PeerRealm" : "example.com",
      "PeerIpAddress" : "127.0.0.1", "PeerPort" : "46397", "TransportProtocol" : "TCP"
    }
    ,
    { "DiameterGatewayName" : "diameterGateway2", "DiameterGatewayRealm" :
      "example.com", "DiameterGatewayIpAddress" : "127.0.0.1", "DiameterGatewayPort" :
      "3869", "PeerName" : "ccr.seagull.client.dgw2.peer2", "PeerRealm" : "example.com",
      "PeerIpAddress" : "10.148.206.194", "PeerPort" : "58006", "TransportProtocol" :
      "SCTP"
    }
    ,
    { "DiameterGatewayName" : "diameterGateway2", "DiameterGatewayRealm" :
      "example.com", "DiameterGatewayIpAddress" : "127.0.0.1", "DiameterGatewayPort" :
      "3869", "PeerName" : "ccr.seagull.client.dgw2.peer1", "PeerRealm" : "example.com",
      "PeerIpAddress" : "127.0.0.1", "PeerPort" : "47145", "TransportProtocol" : "TCP"
    }
  ]
  ...
} ]

```

You can also view the details of the diameter peers that are connected to specific Diameter Gateway instances by using ECE MBeans. See ["Viewing Active Diameter Peers"](#).

Viewing Active Diameter Peers

To view all the active diameter peers:

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 a Diameter Gateway instance 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 **DiameterGateway** node.
2. Expand **PeerConnectionsTracker**.
 3. Expand **Attributes**.
 4. Click the **peerConnections** value.

The diameter peers that are active (which are currently connected to the specific Diameter Gateway instance) are displayed.

For example, if the Diameter peers, **peer1** and **peer2**, are connected to **diameterGateway1**, the following details are displayed:

```
ccr.seagull.client.dgw1.peer2=
{DiameterGatewayName=diameterGateway1, DiameterGatewayRealm=example.com,
DiameterGatewayIpAddress=127.0.0.1, DiameterGatewayPort=3868,
PeerName=ccr.seagull.client.dgw1.peer2, PeerRealm=example.com,
PeerIpAddress=127.0.0.1, PeerPort=52247, TransportProtocol=SCTP}
ccr.seagull.client.dgw1.peer1=
{DiameterGatewayName=diameterGateway1, DiameterGatewayRealm=example.com,
DiameterGatewayIpAddress=127.0.0.1, DiameterGatewayPort=3868,
PeerName=ccr.seagull.client.dgw1.peer1, PeerRealm=example.com,
PeerIpAddress=127.0.0.1, PeerPort=46397, TransportProtocol=TCP}
```

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 Release Notes*.

High Availability Support for ECE Components

In the previous releases, you could not start multiple instances of an ECE component at a time.

With this enhancement, you can configure and start multiple additional instances of the following components in the same system or different ECE systems (such as backup machines) to ensure high availability:

- Rated Event Formatter
- BRM Gateway
- Customer Updater
- Pricing Updater

All the instances of the component configured for high availability can run at the same time with one instance always in the active mode and other instances in the standby

mode. When the active instance of the component goes down due to system failure, one of the instances in the standby mode automatically becomes active and the instance that failed becomes standby. All the other instances continue to remain in the standby mode. You can configure additional instances of these components by using ECE Mbeans and ECE topology settings.

To configure additional instances of these ECE components, see the following:

- For Rated Event Formatter, see the discussion about installing and configuring Rated Event Formatter in *BRM Elastic Charging Engine Implementation Guide*.
- [Configuring Customer Updater Instances](#)
- [Configuring BRM Gateway Instances](#)
- [Configuring Pricing Updater Instances](#)

You can monitor the status of each component and server configured for high availability by using ECE Mbeans. For more information, see "[Viewing ECE High Availability Status](#)".

Configuring Customer Updater Instances

To configure a Customer Updater instance:

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.
2. Expand **charging.connectionConfigurations**.
3. Expand **Operations**.
4. Click **addoracleQueueConnectionConfiguration**.
5. Set the **name** attribute to the name of the additional Customer Updater instance; for example, **CustomerUpdater2**.
6. Open the *ECE_home/occeserver/config/eceTopology.conf* file.
7. Add a row to the file for the Customer Updater instance.

Use a unique name for each instance so that Elastic Charging Controller (ECC) can distinguish between them. (All Customer Updater instances use the same role.)

For example, to configure three Customer Updater instances, add three rows in which the **name** value is **customerUpdater**, **customerUpdater2**, and **customerUpdater3** respectively, and the **role** value for all three rows is **customerUpdater**.

8. Save and close the file.
9. Repeat step 1.
10. Expand **charging.connectionConfigurations.customerUpdater n** , where n is the number that represents the additional instance; for example, **CustomerUpdater2**.

11. Expand **Attributes**.
12. Specify the attributes for the additional Customer Updater instance.

Note: You can configure multiple Customer Updater instances to connect to the same BRM schema.

For information on the attributes, see the discussion about configuring Customer Updater in *BRM Elastic Charging Engine Implementation Guide*.

Configuring BRM Gateway Instances

To configure a BRM Gateway instance:

1. Open the *ECE_home/occeserver/config/eceTopology.conf* file.
2. Add a row to the file for the BRM Gateway instance.

Use a unique name for each instance so that Elastic Charging Controller (ECC) can distinguish between them. (All BRM Gateway instances use the same role.)

For example, to configure three BRM Gateway instances, add three rows in which the **name** value is **brmGateway**, **brmGateway2**, and **brmGateway3** respectively, and the **role** value for all three rows is **brmGateway**.

3. Save and close the file.
4. Specify the attributes for the additional BRM Gateway instance by using ECE Mbeans. For instructions, see the discussion about configuring BRM Gateway in *BRM Elastic Charging Engine Implementation Guide*.

Configuring Pricing Updater Instances

To configure a Pricing Updater instance:

1. Open the *ECE_home/occeserver/config/eceTopology.conf* file.
2. Add a row to the file for the Pricing Updater instance.

Use a unique name for each instance so that Elastic Charging Controller (ECC) can distinguish between them. (All Pricing Updater instances use the same role.)

For example, to configure three Pricing Updater instances, add three rows in which the **name** value is **pricingUpdater**, **pricingUpdater2**, and **pricingUpdater3** respectively, and the **role** value for all three rows is **pricingUpdater**.

3. Save and close the file.

Viewing ECE High Availability Status

To view the ECE high availability status:

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 Monitoring** node.
2. Expand **HighAvailability**.
3. Expand **Attributes**.

The **ActiveInstances** attribute displays the active instance of each component configured for high availability.

The **IsActive** attribute displays **true** or **false** to indicate whether the connected instance is an active instance or a passive instance. The **IsActive** value is displayed only for the components or servers configured for high availability.

The **PassiveInstances** attribute displays all the passive instances of each component configured for high availability.

Improved BRM-to-ECE Data Synchronization

In the previous releases, sometimes there was a lag between the BRM database and the ECE cache update when the BRM data commit failed. During the lag, the BRM and ECE data were unsynchronized.

With this enhancement, the following have been introduced in ECE and BRM:

- The **brmPostCommitEnabled** attribute in ECE specifies whether to commit the data in ECE after the data is successfully committed in BRM. The default value is **false**. You must set this entry to **true** to enable post commit in ECE.

When post commit is enabled, ECE locks the data transaction for the customer for which the update request is received from BRM until the update is committed successfully in BRM. The update requests received for the customer during the lock period fail with the **CUSTOMER_IN_TRANSACTION** error. All the update requests for that customer are processed only after the update is committed successfully in BRM.

- The **updateRequestServerTimeout** attribute in ECE specifies the number of milliseconds in which the update requests from BRM must be processed before EM Gateway times out. The default value is **20000**. The updates requests from BRM are not processed if EM Gateway times out.
- The **ece_post_commit_enabled** entry in BRM specifies to commit the data in the ECE cache after the BRM data commit. The default value is **0**. You must set this entry to **1** to enable post commit in BRM.

You can specify these attributes to ensure that the data in BRM and ECE are synchronized even if the BRM data commit fails. And, if the BRM data commit fails, you must update the customer data by loading it incrementally from BRM into ECE. For more information, see the discussion about loading customer data incrementally with **customerLoader** in *BRM Elastic Charging Engine Implementation Guide*.

To enable post commit:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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, where *ECE_home* is the directory in which ECE is installed.

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 `brmPostCommitEnabled` attribute to **true**.
5. Ensure that real-time synchronization of BRM and ECE data is enabled.

See the discussion about enabling real-time synchronization of BRM and ECE customer data updates in *BRM Elastic Charging Engine Implementation Guide*.

6. Open the `BRM_home/sys/cm/pin.conf` file in a text editor.
7. Add the following entries to the end of the file:

```
-cm em_group ece PCM_OP_ECE_POST_COMMIT
-cm ece_post_commit_enabled 1
```

8. Save and close the file.
9. Restart the CM.

See the discussion about starting and stopping the BRM system in *BRM System Administrator's Guide*.

Summary of Fixes in ECE 11.3 Patch Set 9

Table 1–2 lists the bugs that were fixed in ECE 11.3 Patch Set 9 and provides a brief description of the resolution.

Table 1–2 Bug Fixes in ECE 11.3 Patch Set 9

SR Number	Bug Number	Description
3-17401670661	28082434	ECE displayed errors with null pointer exceptions when the billing process was in progress. This has been fixed.
3-17277297191	28140114	When there were many JMX-enabled server nodes in the ECE cluster, Elastic Charging Controller (ECC) was taking considerable time to start the server nodes. This has been fixed.
3-17672497271	28273069	Rating was failing due to the incorrect balance retrieved from the alteration agreements. This has been fixed.
3-16869958441	28289482	The connection retry configuration for Pricing Updater was inconsistent compared to other similar processes. This has been fixed.
3-17190685401	28289485	Pricing Updater was failing with an error due to unicode characters. This has been fixed.
3-17248648931	28290158	During usage processing, ECE was displaying a null pointer exception. This has been fixed.

Table 1–2 (Cont.) Bug Fixes in ECE 11.3 Patch Set 9

SR Number	Bug Number	Description
3-17686847261	28292700	There was an increase in the CPU utilization when Customer Loader failed to load customer data into the ECE cache. This has been fixed.
3-17820829771	28319315	ECE stopped processing requests due to a deadlock encountered while accessing the recurring bundle history. This has been fixed.
3-16899337801	28331673	There was an issue observed in the service transfer when the payload to EM gateway did not include certain mandatory parameters. This has been fixed.
3-17223827112	28359602	Sometimes there was a lag between the BRM database and the ECE cache update when the BRM data commit failed. This has been fixed.
3-17677242851	28448499	The ECE Monitoring Agent did not log the Diameter messages in the Diameter Gateway statistics report. This has been fixed.
3-17518467721	28485144	Diameter Gateway was returning the success result though the corresponding user ID was not present in the identity repository. This has been fixed.
3-17704998311	28486837	The pre-rating extension returned a null pointer exception when using the getSharedContext method in the extension context. This has been fixed.
3-17912030531	28492859	When the customer data was deleted from the ECE cache, the balance information was not deleted if it was not associated at the product level or if it was not the default balance. This has been fixed.
3-17993806241	28504037	The Diameter Gateway response did not include the Multiple-Services-Credit-Control (MSCC) block and this resulted in an intermittent null pointer exception. This has been fixed.
3-18007689131	28537495	The notifications were not getting published to the WebLogic JMS queue. This has been fixed.
3-18067230701	28544884	When a patch set was installed in a disaster recovery deployment, the <code>ECE_home/occeserver/config/management/charging-setting s.xml</code> file in the secondary site was not automatically updated with the latest configurations. This has been fixed.
3-16175651581	28569962	The <code>PCM_OP_SUBSCRIPTION_SHARING_GROUP_SET_PARENT</code> opcode was failing in ECE if the parent account did not have any chargeshare offer. This has been fixed.

Table 1–2 (Cont.) Bug Fixes in ECE 11.3 Patch Set 9

SR Number	Bug Number	Description
3-18087254131	28580491	During failover, BRM Gateway could not reconnect to the WebLogic JMS queue. This has been fixed.
3-17997864061	28621056	ECE did not update the ECE cache if a customer service representative (CSR) account was closed in BRM using the PCM_OP_CUST_SET_STATUS opcode. This has been fixed.
3-17272033681 3-17266994221	27963061 28124268	After top-up, rating was failing with the credit floor breach error. This has been fixed.
3-18036906711	28642033 28649377	In ECE, some performance issues were caused by the parse method of the TimeUtil class. This has been fixed.

Known Problems in ECE 11.3 Patch Set 9

This section provides an overview of the known problems in ECE 11.3 Patch Set 9.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)
- [Pricing Updater Configured for High Availability Is Not Working As Expected](#)

Pricing Updater Configured for High Availability Is Not Working As Expected

SR Number: Not applicable

Bug Number: 28470833

When the active instance of Pricing Updater goes down while processing the pricing data, the remaining data in the queue is not processed by the standby instance and the data is lost.

To work around this problem, do the following in PDC:

Publish all the PDC pricing data (the metadata, setup, and pricing data) from the PDC database to ECE by running the following commands:

```
ImportExportPricing -publish metadata -target [ece]
ImportExportPricing -publish config -target [ece]
ImportExportPricing -publish pricing -target [ece]
```

This ensures that all the pricing data are processed by the currently active instance of Pricing Updater and are published to ECE.

ECE 11.3 Patch Set 8 Release Notes

This section provides information about ECE 11.3 Patch Set 8.

New Features in ECE 11.3 Patch Set 8

This section provides documentation for the features introduced in ECE 11.3 Patch Set 8.

ECE Now Sends Group Notifications to Subscribers

In previous releases, ECE was sending the credit limit and threshold breach notifications only to individual subscribers. For example, in case of sharing groups, when a credit threshold was breached, ECE sent the threshold breach notification only to the individual subscriber associated with the breach and not to all members in that sharing group.

With this enhancement, you can enable ECE to send group notifications for the following type of notifications:

- Threshold Breach Notifications
- Credit Ceiling Breach Notifications
- Credit Floor Breach Notifications

You can perform this by setting the **groupNotificationEnabled** entry in the *ECE_home/occeserver/config/management/charging-settings.xml* file. For more information, see ["Enabling Group Notifications"](#).

By default, the group notifications feature is disabled and ECE sends the breach notifications only to the corresponding individual subscribers.

When you enable the group notifications feature, ECE sends the group notifications to specific subscribers or a group based on the subscriber preferences data received from BRM. You can manage how each subscriber prefers to receive notifications from the network by customizing subscriber preferences in BRM. For example, if a subscriber is a member of multiple sharing groups and wants to receive breach notifications only from a particular group, you can specify this preference by customizing the subscriber profile data configuration. If the subscriber is an owner of the sharing group, the breach notifications are sent to the subscriber irrespective of the preference set in BRM. For more information, see the discussion about subscriber preferences in *BRM Telco Integration*.

To enable group notifications for specific subscribers, you must add the **NotificationEnabledAgreements** subscriber preference in BRM with the list of subscribers for sending group notifications. You can change this preference based on the requirements. For more information, see ["Enabling Group Notifications for Specific Subscribers"](#).

Enabling Group Notifications

To enable group notifications:

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.
2. Expand **charging.server**.
3. Expand **Attributes**.
4. Set the **groupNotificationEnabled** attribute to **True**.

Enabling Group Notifications for Specific Subscribers

Important: Before enabling group notifications, do one of the following if you want to enable group notifications for existing subscribers (sharing group owners):

- Load customer data incrementally into ECE. For more information, see the discussion about incremental loading of customer data in *BRM Elastic Charging Engine Implementation Guide*.
 - Restart ECE. For information about starting and stopping ECE, see *BRM Elastic Charging Engine System Administrator's Guide*.
-

To enable group notifications for specific subscribers:

1. Add a string for the **NotificationEnabledAgreements** preference in the *BRM_home/sys/messages/active_mediation/active_mediation.en_US* file. For example:

```
STR
  ID = 8 ;
  Version = 1 ;
  STRING = "NotificationEnabledAgreements" ;
END
```

For instructions, see the discussion about localizing and customizing strings in *BRM Developer's Guide*.

2. Add the **NotificationEnabledAgreements** preference in the *BRM_home/sys/data/config/config_subscriber_preferences_map.xml* file. For example:

```
</SUBSCRIBER_PREFERENCES>
  <SUBSCRIBER_PREFERENCES elem="7">
    <NAME>NotificationEnabledAgreements</NAME>
    <SUBSCRIBER_PREFERENCE_ID>8</SUBSCRIBER_PREFERENCE_ID>
    <STRING_ID>8</STRING_ID>
    <STR_VERSION>1</STR_VERSION>
    <DEFAULT></DEFAULT>
    <TYPE>1</TYPE>
  </SUBSCRIBER_PREFERENCES>
```

For instructions, see the discussion about customizing subscriber preferences in *BRM Telco Integration*.

3. For the **NotificationEnabledAgreements** preference, specify the list of sharing group names of the subscribers (for example, SharingAgreement12, Charge_Sharing14, DataSharing_143) as a comma-separated value.

You can set the **NotificationEnabledAgreements** preference value by using:

- The `PCM_OP_CUST_SET_SUBSCRIBER_PREFERENCES` opcode. For more information, see the discussion about the `PCM_OP_CUST_SET_SUBSCRIBER_PREFERENCES` opcode in the *BRM Developer's Guide*.
- Customer Center. For more information, see the discussion about setting the subscriber preference in the Customer Center Online Help.

Diameter Gateway Configuration Now Includes SCTP Attributes

In previous releases, to use Stream Control Transmission Protocol (SCTP) for network communication, you had to configure SCTP at the operating system (OS) level.

ECE now contains the attributes for configuring SCTP. You can configure SCTP when you configure the Diameter Gateway nodes in ECE. For more information, see ["Configuring SCTP"](#).

Configuring SCTP

To configure SCTP:

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.
2. Expand **charging.diameterGatewayConfigurations.Instance_Name**, where *Instance_Name* is the name of the instance to configure.
3. Expand **Attributes**.
4. Specify values for the following attributes:
 - **sctpMaxInStream**. Specify the maximum number of incoming streams allowed per association. Valid range is 0 to 65535.
 - **sctpMaxOutStream**. Specify the maximum number of outgoing streams allowed per association. Valid range is 0 to 65535.
 - **sctpReceiveBufferSize**. Specify the buffer size for receiving data (in bytes). Valid range is 0 to 2147483647.
This attribute determines the SCTP receiver window size. You can increase or decrease the size based on your requirement. For example, you can increase the size for high-volume connections or decrease the size if you want to limit the incoming data backlog.
 - **sctpSendBufferSize**. Specify the buffer size for sending data (in bytes). Valid range is 0 to 2147483647.
This attribute determines the maximum amount of data that can be sent in a single transaction. You can increase or decrease the size based on your requirement.
5. Change directory to the *ECE_home/occeserver/bin* directory.
6. Start Elastic Charging Controller (ECC):


```
./ecc
```
7. Do one of the following:
 - If the Diameter Gateway instance is *not* running, start it.
The instance reads its configuration information by name at startup.
 - If the Diameter Gateway instance is running, stop and restart it.

For information about stopping and starting Diameter Gateway instances, see the discussion about starting and stopping ECE in *BRM Elastic Charging Engine System Administrator's Guide*.

Diameter Gateway Now Supports Both Incremental and Cumulative Accounting

ECE supports both incremental-based and cumulative-based accounting when processing usage requests. However, in previous releases, Diameter Gateway was supporting only the incremental-based accounting.

With this enhancement, you can configure Diameter Gateway to use both incremental-based and cumulative-based accounting logic when processing usage requests. You can perform this by specifying the accounting mode in the mediation specification file. The accounting mode indicates to Diameter Gateway whether to use incremental-based or cumulative-based accounting logic. For more information, see "[Configuring Accounting Mode for Diameter Gateway](#)".

Configuring Accounting Mode for Diameter Gateway

To configure the accounting mode for Diameter Gateway:

1. Open the Diameter mediation specification file, **diameter_mediation.spec**.

For the location of the **diameter_mediation.spec** file, see the **configObjectsDataDirectory** parameter in the *ECE_home/occeserver/config/management/migration-configuration.xml* file.

2. For each event to be rated (in each row), specify the accounting mode in the **UnitReportingMode** column. Valid values are:

- INCREMENTAL
- CUMULATIVE

For example:

```
Service-Context-Id | Service-Identifier | Rating-Group | ProductType |
EventType | Version | Subscription-Id-Type | ValidFrom | UnitReportingMode |
"gy.service@example.com" | "1" | "10" | "VOICE" | "V_USAGE" | 1.0 | "" |
"2014-3-31T12:01:01 PST" | "INCREMENTAL" |
"gy.service@example.com" | "1" | "11" | "DATA" | "D_USAGE" | 1.0 |
"" | "2014-3-31T12:01:01 PST" | "CUMULATIVE" |
```

For more information on incremental and cumulative accounting, see the discussion about using incremental or cumulative accounting for usage requests in *BRM Elastic Charging Engine Implementation Guide*.

The default accounting mode is **Incremental**. If you specify null or if you do not specify a mode in the **UnitReportingMode** column, Diameter Gateway uses the default accounting mode when processing usage requests. This supports backward compatibility.

Your accounting mode configuration is applicable for the combination of the given values specified in the row for the event to be rated. You can also configure different accounting modes for the same product and event type combination.

3. Save and close the file.
4. Run the **configLoader** utility to load your mediation specification in the ECE cluster:

```
start configLoader
```

When the mediation specification is loaded, the earlier version of your mediation specification (that was in the ECE cluster) is overwritten and Diameter Gateway uses the configuration of the newly loaded mediation specification.

5. Change directory to the `ECE_home/occeserver/bin` directory.

6. Start ECC:

```
./ecc
```

7. Do one of the following:

- If the Diameter Gateway instance is *not* running, start it.

The instance reads its configuration information by name at startup.

- If the Diameter Gateway instance is running, stop and restart it.

For information about stopping and starting Diameter Gateway instances, see the discussion about starting and stopping ECE in *BRM Elastic Charging Engine System Administrator's Guide*.

Inactive and Dormant Accounts are Now Purged from the ECE Cache

In previous releases, deleting a customer account (including prepaid calling cards) in BRM was not purging the account from the ECE cache. The deleted account still remained in the ECE cache affecting the memory usage.

To optimize and efficiently manage the memory, ECE now allows you to purge inactive and dormant accounts from the ECE cache. You can purge the accounts synchronously from both BRM and ECE in real time by using EM Gateway.

When you delete an account from the BRM database by using the `PCM_OP_CUST_DELETE_ACCT` opcode, BRM sends the update request to ECE through EM Gateway. ECE processes the update request and purges the account from the ECE cache if there are no active sessions associated with that account. The BRM database and the ECE cache are updated within the same transaction in real time. This ensures that the BRM and ECE data remain synchronized.

When purging the account from the ECE cache, ECE purges the customer data associated with the account, which includes:

- Customer information
- Customer balance history
- Recurring bundle history
- Account top-up history
- Sharing customer state history
- Billing trigger cycle information
- Public user identity (PUID), if it is not associated with any other active account

To purge the accounts synchronously on BRM and ECE, you must enable synchronous updates on BRM and ECE. For more information, see the discussion about enabling real-time synchronization of BRM and ECE customer data updates in *BRM Elastic Charging Engine Implementation Guide*.

Post-Rating Extension Can Now be Used to Add or Delete Rating Periods

You can now implement a custom logic to add or delete rating periods after rating by using the post-rating extension. For more information on using the post-rating extension, see *ECE Extensions*.

See the **PostRatingConsolidateRatingPeriods** sample program in the ECE SDK (*ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions* directory) for an example. This sample program shows how to use the ECE extensions API to:

- Add a single rating period with the consolidated charge for all the rating periods of type CHARGE.
- Delete all the existing rating periods of type CHARGE.

You can use this sample program to access the ECE cache and override the rating periods in the final rated results by adding or deleting rating periods.

Summary of Fixes in ECE 11.3 Patch Set 8

Table 1–5 lists the bugs that were fixed in ECE 11.3 Patch Set 8 and provides a brief description of the resolution.

Table 1–3 Bug Fixes in ECE 11.3 Patch Set 8

SR Number	Bug Number	Description
3-16341238331	27222381	When an account was deleted in BRM, the deleted account and the corresponding data was not purged from the ECE cache. This has been fixed.
3-15274947261	27558167	In a high-availability (HA) system, after performing the rolling upgrade, the HA status was not properly reset to NODE-SAFE. This has been fixed.
3-17023814901	27672184	Discounts were not evaluated based on the start time of a session. This has been fixed.
3-17003480611	27673055	The timeout exception in EM Gateway was causing issues in real-time synchronization between ECE and BRM. This has been fixed.
3-16906316991	27710068	The rounding function was incorrectly rounding when using the post-rating extension. This has been fixed.
3-17120179241	27733370	When the CohQL query tool was stopped manually using the Quit command in normal circumstances, ECE was unnecessarily logging an error. This has been fixed.
3-17084672084	27738519	JMS connection issues were logged as debug messages instead of warning messages. This has been fixed.
3-16752555261	27763552	The EBAL functionality was not supported in ECE for aggregated distribution processing. This has been fixed.

Table 1–3 (Cont.) Bug Fixes in ECE 11.3 Patch Set 8

SR Number	Bug Number	Description
3-16377347041	27859874	EM Gateway was taking a longer time to start. This has been fixed.
3-17260603111	27910681	After reloading the customer into the ECE cache, ECE was unnecessarily creating another item for the bill in progress. This has been fixed.
3-17301488851	27917750	When a login was disassociated in BRM, the same was not disassociated in ECE. This has been fixed.
3-16641250471	27936536	The current balance was an exponential value even though it was configured to be a decimal value with a specific precision. This has been fixed.
3-17008041041	27968411	There were issues with the handling of zero duration sessions. This has been fixed.
3-16196745521	27974153	There was an issue in updating the correct life cycle state of subscribers when the ECE extensions were used. This has been fixed.
3-17366629941	28034359	Modifying or setting tax rate was not available in the ECE extensions workflow. This has been fixed.
3-17530769731	28076346	The PCM_OP_BAL_CHANGE_VALIDITY opcode was not synchronizing the validity information of resources when called during the accounting workflow. This has been fixed.
3-17346180211	28101305	Whenever the redirection rule was triggered, if the next update request did not include the Reporting-Reason value, ECE was returning Final Unit Indicator (FUI) and Granted-Service-Unit (GSU) instead of Validity-Time. This has been fixed.
3-16690428221	28141462	When an account was moved from the PRE-ACTIVE or ACTIVE state, it was not possible to detect the transition in Diameter Gateway or ECE extensions for any further customization. This has been fixed.

Known Problems in ECE 11.3 Patch Set 8

This section provides an overview of the known problems in ECE 11.3 Patch Set 8.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)

ECE 11.3 Patch Set 7 Release Notes

This section provides information about ECE 11.3 Patch Set 7.

New Features in ECE 11.3 Patch Set 7

This section provides documentation for the features introduced in ECE 11.3 Patch Set 7.

ECE Now Supports Incremental Rating for Tariff Changes

ECE now performs incremental rating of an active network session based on the tariff changes that occur during the session. You can configure ECE to generate a rated event whenever a tariff change occurs during a network session. The tariff change can include: peak and off-peak rate changes in offers, availability or expiry of charge offers, alteration offers (discount offer) or distribution offers (charge sharing offers), availability or expiry of customer balances. The incremental rating for tariff changes enables ECE to rate long network sessions incrementally based on the exact data consumed between tariff changes. It also enables operators to show subscribers the running balance based on the actual data consumption after each tariff change.

You can enable incremental rating for tariff changes by configuring the following entries in the `ECE_home/occeserver/config/management/charging-settings.xml` file, where `ECE_home` is the directory in which ECE is installed:

- **enabledOrDisableNonLinear.** For configuring this entry, see ["ECE Now Supports Incremental Rating for Mid-session Rating Condition Changes"](#).
- **tariffTimeChangeSupported.** For configuring this entry, see ["Configuring Tariff Time Change"](#).

When this feature is enabled, ECE determines if there is a tariff change when authorizing and reserving a balance for a session request from the network. ECE bases the reservation on the requested service units of the session request and sends the Tariff-Time-Change AVP in the usage response to the network to record the exact data consumed before and after the tariff change. ECE also performs the reverse rating to calculate the amount of usage that the subscriber can afford and reserves the balance for the requested service units based on the worst-case charging condition (the maximum charge that can be applied for the requested service units). This ensures that the overall usage does not exceed the credit limit of a customer and there is no revenue leakage whether the balance is consumed before or after the tariff change. For more information on reverse rating, see *BRM Elastic Charging Engine Concepts*.

For incremental rating, ECE supports only one tariff change for a session request. If more than one tariff change is determined during authorization, ECE considers the tariff change that occurs first for balance reservation and adjusts the validity time to expire at the next tariff change. This ensures that there is only one tariff change that occurs before the session expires. ECE then rates the exact balance consumed based on the rating condition changes before and after the tariff change and generates rated events each time a tariff change occurs in an ongoing session.

The following section describes a scenario of tariff change during a data session and the balances and charges applied during and at the end of the session.

Scenario: Offer Becomes Valid and Consumed in a Session

Given a subscriber with:

- A charge offer named *CO Data Standard* with a *Standard Bytes* (Chargeable Balance) followed by *Free Use Data* (Counter)
- A starting balance of 50 megabyte (MB) *Standard Bytes* and 0 Bytes *Free Use Data*
- A discount offer named *Data Boost* which is valid from 1:30 PM the same day (100 MB *Standard Bytes* valid from 1:30 PM for 30 minutes from the first use).

The usage charging flow is as follows:

- When the subscriber starts a data session at 1.00 PM
- Consumes 50 MB over 20 minutes at high QoS
- Consumes 2 MB over 10 minutes at low QoS
- The purchased *Data Boost* discount offer becomes valid at 1:30 PM
- Consumes 100 MB over 20 minutes at high QoS
- Consumes 5 MB over 15 min at low QoS before terminating the session

Subscriber's balances and charges are calculated as follows:

Table 1–4 Subscriber Balance and Charge Details

Balance Name	Balance Value	Units Charged
Data Standard	0 MB	150 MB
Free Use Data	7 MB	7 MB

Configuring Tariff Time Change

To configure tariff time change:

Note: You can configure the tariff time change as a system-wide setting and it is applied irrespective of the product type. When configured, the tariff time change is considered for all the products for which incremental rating is enabled.

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand **charging.server**.
3. Expand **Attributes**.
4. Select the **tariffTimeChangeSupported** attribute and set the value to **true**.

ECE Now Supports Dynamic Quota Allocation for Parallel Sessions

In previous releases, if the network did not request for specific reserved units, ECE was allocating the same quota for all the parallel sessions of a subscriber irrespective of the consumption rate. This had an impact on the sessions that require more quota. When multiple parallel sessions approached the charging or policy thresholds, the quota could not be redistributed effectively if there was insufficient balance. Also, this quota allocation was causing an increase in the network signaling traffic due to frequent reauthorization requests.

With this enhancement, ECE supports dynamic quota. Dynamic quota allows you to allocate the available quota dynamically for each parallel session of a subscriber based on the rules you configure in PDC. This adds flexibility in quota allocation and enables the subscribers to run concurrent online charging sessions effortlessly. This also enables you to optimize the network usage effectively.

You define the rules for determining the dynamic quota allocation by configuring dynamic quota selectors in PDC. These rules are expressions that reference any of, or a combination of, the following attributes:

- Event
- Customer
- Service
- Profile attribute specifications

You configure the dynamic quota selector for each service-event combination and then publish them to ECE. For more information on configuring dynamic quota selectors, see "Dynamic Quota Configuration is Now Supported in PDC (Patch Set 12)" in *PDC Release Notes*.

When ECE receives the usage requests from the network in which the requested service unit (RSU) is not set, it evaluates and applies the rules from the dynamic quota selectors on the usage request to derive the quota to be allocated and the quota attributes, such as the quota holding time (QHT), the volume quota threshold (VQT), and the quota validity time (VT). If dynamic quota selector rules are not configured for a service-event combination, ECE uses the default quota configuration for deriving the quota and the quota attributes. ECE returns the derived quota (as granted service unit (GSU)), QHT, and VQT values in the usage response to the network.

If you want to customize the dynamic quota allocation to suit your business requirements, you can modify the RSU and the quota attribute values in the usage request by using the pre-rating extension. For more information on modifying RSU, see "[Pre-Rating Extension Enhanced to Support Dynamic Quota](#)". For more information on modifying the quota attribute values in the usage request, see "[Additional Attributes for Reservation Quota Management](#)".

You can also modify the QHT and VQT values in the usage response by using the post-charging extension. For more information on modifying the QHT and VQT values in the usage response, see "[Additional Attributes for Reservation Quota Management](#)".

In addition, if you want to configure large quotas and long reservation periods to reduce network signaling, you can trigger server-initiated reauthorization requests (RARs) to get a subscriber's exact balance usage before performing other business operations. For more information, see "[ECE Extensions Can be Used to Trigger RAR Notifications for Ongoing Sessions](#)".

ECE Extensions Can be Used to Trigger RAR Notifications for Ongoing Sessions

When you use dynamic quotas for long running sessions to reduce network signaling, you can trigger server-initiated reauthorization requests to get the exact reservation balance before performing other business operations.

To generate server-initiated reauthorization requests, you must generate RAR notifications. To generate these notifications, you can now implement custom logic by using the following ECE extensions in the rating flow:

- Pre-rating extension

- Post-rating extension
- Post-charging extension

You can use these extensions to access the ECE cache and generate RAR notifications. In the custom logic, if the **SendGenericRARNotification** is set to **true**, ECE generates generic RAR notifications for all Diameter sessions for the client and the Rating-Group and Service-Identifier are not set in those notifications. If

SendGenericRARNotification is set to **false**, ECE generates service-specific RAR notifications with Rating-Group and Service-Identifier set in the notifications.

See the following sample programs in the *ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions* directory for the example on how to implement the custom logic by using the pre-rating, post-rating, and post-charging extensions:

- **SampleRarPreRatingExtension**
- **SampleRarPostRatingExtension**
- **SampleRarPostChargingExtension**

ECE Now Monitors Diameter Statistics

The ECE Monitoring Agent monitors the main components of ECE, such as ECE cluster, and ECE caches. In addition, the Monitoring Agent now monitors the Diameter sessions and responses at regular intervals and logs the status of the sessions in the **monitorAgentX_ECE_SUMMARY_REPORT.log** file. By default, this log file is stored in the *ECE_home/occeserver/logs* directory. For more information on the Monitoring Agent, see "[Support for Monitoring ECE](#)".

The Monitoring Agent monitors the following Diameter sessions and responses:

- **Charging sessions.** Monitors the status of all charging sessions and logs the total number of sessions that are opened or closed after the last monitoring check and the sessions that are currently active at the time of current monitoring check. This summary includes the overall status of the charging sessions.
- **Network sessions.** Monitors the status of all network sessions and logs the total number of sessions that are opened or closed after the last monitoring check and the sessions that are currently active at the time of the current monitoring check. This summary includes the overall status of the network sessions.
- **Diameter Responses.** Monitors all the Gy and Sy responses and their result codes and logs the total number of success and failure responses. This summary includes the overall status of the responses and the details of the failed responses for each result code and product type. The product type is a combination of Service-Context-Id, Service-Identifier, and Rating-Group and it is shown in the summary only for the Initiate, Update, Terminate (IUT) requests. For other requests, such as TopUp, Debit, and Refund, the product type is not applicable and it is shown as **Unknown**.

The following is an example of the summary log generated for the Diameter sessions:

```
"DiameterGatewayStatistics" : {
...
  "Session" : {
    "Charging" : {
      "PerRatingGroup" : [ {
        "RatingGroup" : 10,
        "Open" : 506,
        "Active" : 1011,
```

```
        "Close" : 505
      }, {
        "RatingGroup" : 15,
        "Open" : 1000,
        "Active" : 980,
        "Close" : 20
      } ],
      "Summary" : {
        "Open" : 1506,
        "Active" : 1991,
        "Close" : 525
      }
    },
    "Network" : {
      "Close" : 5,
      "Active" : 3251,
      "Open" : 3193
    }
  }
},
"ResponseStatus" : {
  "Gy" : {
    "FailureDetails" : {
      "PerResultCode" : {
        "Individual" : [ {
          "ResultCode" : 5030,
          "Count" : 1
        }, {
          "ResultCode" : 5012,
          "Count" : 1
        } ],
      "MSCC" : [ {
        "ResultCode" : 5030,
        "Count" : 1
      }, {
        "ResultCode" : 5012,
        "Count" : 1
      } ]
    },
    "PerProductType" : [ {
      "ProductType" : "32260@3gpp.org+null+10",
      "Count" : 2
    }, {
      "ProductType" : "32260@3gpp.org+1+10",
      "Count" : 2
    } ]
  },
  "Success" : 5,
  "Failure" : 4
},
"Sy" : {
  "FailurePerResultCode" : [ {
    "ResultCode" : 5030,
    "Count" : 1
  }, {
    "ResultCode" : 5570,
    "Count" : 2
  }, {
    "ResultCode" : 5004,
    "Count" : 1
  } ]
}
```

```

    } ],
    "Success" : 5,
    "Failure" : 4
  }
}
}

```

BRM Gateway Now Automatically Processes the Failed Update Requests

In previous releases, you had to manually move the failed update requests from the BRM Gateway suspense queue to BRM Gateway for reprocessing those requests.

With this enhancement, ECE automatically moves the failed update requests from the BRM Gateway suspense queue to BRM Gateway at regular intervals. You can change the interval at which the requests are moved from the BRM Gateway suspense queue and the maximum number of requests that can be moved to BRM Gateway at a time by setting the **brmSuspenseQueuePeriod** and **jmsBatchSize** entries (respectively) in the *ECE_home/occeserver/config/management/charging-settings.xml* file. See ["Modifying the Interval and Batch Size for Moving Failed Update Requests to BRM Gateway"](#).

ECE moves the update requests failed due to network failure or other exceptions in BRM; for example, credit limit breach. However, if the update request failed due to incorrect message format, you must manually fix the format and then move the update request to BRM Gateway. You can perform this by using WebLogic Server Administration Console.

Before fixing the message format, you can increase the interval for moving the failed update requests to BRM Gateway, as required, so that the requests are moved to BRM Gateway only after you fix the message format. See ["Modifying the Interval and Batch Size for Moving Failed Update Requests to BRM Gateway"](#).

To fix the message format and move the update request to BRM Gateway, do the following in WebLogic Server Administration Console:

1. Export the update request into an XML file.
For instructions, see the discussion about exporting messages in the WebLogic Server Administration Console Help.
2. Fix the message format by editing the XML file.
3. Import the fixed update request in the XML format.
For instructions, see the discussion about importing messages in the WebLogic Server Administration Console Help.
4. Move the update request to BRM Gateway.
For instructions, see the discussion about removing excess failed updates in the BRM Gateway suspense queue in *BRM Elastic Charging Engine System Administrator's Guide*.

Modifying the Interval and Batch Size for Moving Failed Update Requests to BRM Gateway

To modify the interval and batch size for moving the failed update requests from the BRM Gateway suspense queue to BRM Gateway:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.

- 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.
2. Expand **charging.brmGatewayConfiguration**.
 3. Expand **Attributes**.
 4. Specify the values for the following attributes:
 - **brmSuspenseQueuePeriod**. Specify the interval (in milliseconds) for moving the failed update requests from the BRM Gateway suspense queue to BRM Gateway.
 - **jmsBatchSize**. Specify the maximum number of failed update requests to be moved from BRM Gateway suspense queue to BRM Gateway in a batch.

ECE Now Uses Policy Specifications for Policy-Driven Charging

ECE supports policy-driven charging. For ECE to support policy-driven charging, you need to configure offer profiles with policies to define the ranges in the quality of service (QoS) for the services you offer.

In the previous releases, you had to create the offer profiles in BRM and then load the offer profiles from BRM into ECE.

ECE now uses the policy specifications (offer profiles in BRM) configured in PDC for policy-driven charging. You can configure them directly in PDC and then publish them to ECE by using Pricing Updater.

Note: Ensure that you migrate the offer profile data from BRM into PDC and *do not* create any offer profiles in BRM.

For more information, see the discussion about configuring policy specifications in *PDC Release Notes*.

Summary of Fixes in ECE 11.3 Patch Set 7

[Table 1–5](#) lists the bugs that were fixed in ECE 11.3 Patch Set 7 and provides a brief description of the resolution.

Table 1–5 Bug Fixes in ECE 11.3 Patch Set 7

SR Number	Bug Number	Description
3-14902680971	26076931	<p>When configuring ECE for disaster recovery, two Diameter Gateway nodes could not be configured with the same name; for example, diameterGateway1.</p> <p>This has been fixed by adding a new parameter, clusterName, in the Diameter Gateway configuration. Now, ECE uses both name and clusterName to identify a Diameter Gateway instance. You must specify a different clusterName if you are configuring Diameter Gateway nodes with the same name. For example, you can configure two Diameter Gateway nodes as follows:</p> <p>Diameter Gateway 1</p> <pre>name='diameterGateway1' clusterName="cluster1"</pre> <p>Diameter Gateway 2</p> <pre>name='diameterGateway1' clusterName="cluster2"</pre> <p>Note: Ensure that this clusterName matches the clusterName in the <i>ECE_home/occeserver/config/charging-coherence-override-secure-prod.xml</i> file.</p> <p>For more information, see the discussion about specifying Diameter Gateway node properties in <i>BRM Elastic Charging Engine System Administrator's Guide</i>.</p>
3-14963166661	26451761	<p>ECE was assigning the usage events to newly created bill items instead of using the existing items that were already associated with the service.</p> <p>This has been fixed.</p>
3-15623565771	27010961	<p>Incorrect match factor was used in the discount calculation resulting in incorrect discounts.</p> <p>This has been fixed.</p>
3-16475074081	27365713	<p>Rated Event Formatter was overriding the END_T values in the event records even if the information was passed in the input.</p> <p>This has been fixed.</p>
3-16378591971	27379542	<p>Concurrent time out was observed when Customer Updater was processing events, such as productInfoChange, discountInfoChange and sponsorInfoChange.</p> <p>This has been fixed.</p>
3-15392504701	27387822	<p>When Customer Loader failed with errors, high CPU utilization was observed in the machines on which ECE was running.</p> <p>This has been fixed.</p>
3-16661346291	27412494	<p>The breach direction in the threshold breach notifications was incorrect.</p> <p>This has been fixed.</p>

Table 1–5 (Cont.) Bug Fixes in ECE 11.3 Patch Set 7

SR Number	Bug Number	Description
3-16577311541	27447256	If the invoice data was more than 4000 characters, rating failed. This has been fixed.
3-16507627171	27500490	Even after making a top-up, the session ended with the CREDIT_FLOOR_BREACH error. This has been fixed.
3-16716001101	27500491	When Customer Loader or Customer Updater was run, the validity was incorrectly set to zero (0). As a result, incorrect balances were loaded into ECE. This has been fixed.
3-16628220373	27500492	The virtual time functionality was not working as expected. This has been fixed.
3-16766975451	27500493	The get balance query was failing with a null pointer exception, causing the processing threads to be unresponsive. This has been fixed.
NA	27500494	During the balance update or usage, it was observed that all the rating periods in the rating results were empty in the ECE extension framework. This has been fixed.
3-16752555261	27509047	The EBALs (event_balance_IDs) could not be used for charge distribution. This has been fixed.
3-16392528631	27532751	When a transaction initiated from BRM failed in ECE, the error message was not logged appropriately. This has been fixed.
3-16229758391	27585518	Rated Event Formatter was generating unnecessary rated event files. This has been fixed.
3-16955195711	27628665	When sharing was enabled, the discount was wrongly calculated due to incorrect additional data. This has been fixed.
3-16947888271	27638788	In a multi-RUM scenario with match factor, discounts calculated were incorrect. This has been fixed.
3-16966967681	27669662	For the subscriber tracing functionality, the configuration update to the ECE nodes was always returning the success result even if the update failed. This has been fixed.
3-17003480611	27673055	There were synchronization issues due to incorrect caching scheme. This has been fixed.
3-14767795921	27712668	Customer Updater was hanging while performing an initial load of customers. This has been fixed.

Table 1–5 (Cont.) Bug Fixes in ECE 11.3 Patch Set 7

SR Number	Bug Number	Description
NA	27714212	The match factor was not being considered during the discount split. This has been fixed.
3-16966967681	27717927	In certain cases, ECE was not generating separate logs for subscriber-based tracing. Also, Jconsole was not displaying the errors occurred during the subscriber-based tracing configuration. This has been fixed.
3-16906316991	27719430	When the post-rating extension was used, the <code>unitValue</code> rounding was not working as expected. This has been fixed.

Known Problems in ECE 11.3 Patch Set 7

This section provides an overview of the known problems in ECE 11.3 Patch Set 7.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)

ECE 11.3 Patch Set 6 Release Notes

This section provides information about ECE 11.3 Patch Set 6.

New Features in ECE 11.3 Patch Set 6

This section provides documentation for the features introduced in ECE 11.3 Patch Set 6.

Improved Rating Performance

In the previous releases, when accounts had large and complex rate plans, such as plans with zoning and large number of selectors, the rating performance was affected.

With this enhancement, ECE processes have been enhanced to handle large and complex pricing data. This improves the rating performance for accounts with large and complex rate plans.

Summary of Fixes in ECE 11.3 Patch Set 6

[Table 1–6](#) lists the bugs that were fixed in ECE 11.3 Patch Set 6 and provides a brief description of the resolution.

Table 1–6 Bug Fixes in ECE 11.3 Patch Set 6

SR Number	Bug Number	Description
3-14963166661	26451761	External Manager (EM) Gateway was generating incorrect item lists for UpdateServicesEvent. This has been fixed.
3-16071613021	27098184	For discounts with cascading rules, rules were getting not applied properly. This has been fixed.
3-15931904051	27108893	When a new service is created with a product using the PCM_OP_CUST_MODIFY_CUSTOMER opcode, items were assigned incorrectly. This has been fixed.
3-16135504191	27134194	ECE did not consider the Friends and Family promotion and applied incorrect discounts for the usage events. This has been fixed.
3-16156108231	27280126	The granted service units were calculated incorrectly. This has been fixed.
3-16475074081	27307448	While processing rated events, Rated Event Formatter was overriding the END_T value for the event with a calculated value even if the END_T value was passed in the input. This has been fixed.
3-16401759681	27346547	Close billing was failing with an item list update error. This has been fixed.
3-16280241981	27346900	In a multischema environment, Customer Updater was not loading customer data properly. This has been fixed.
3-16468292701	27356510	The item sponsor information was not getting updated in the customer cache. As a result, ECE was creating new item sponsors while processing the usage of the members in sharing groups. This was causing Rated Event Loader to fail. This has been fixed.
3-16556643741	27359271	When the account-level charge sharing was configured with credit floor notification as None, usage processing failed with an error. This has been fixed.
3-16517334781	27359277	Customer updater was not considering the order specified in the balance group. As a result, the discounts were applied in incorrect order. This has been fixed.
3-16288148591	27359659	The infoCollector command was failing when run on Solaris environment. This has been fixed.
3-16603732751	27360848	Discount sharing was not considered for calculating charges. This resulted in incorrect charges for the events. This has been fixed.

Table 1–6 (Cont.) Bug Fixes in ECE 11.3 Patch Set 6

SR Number	Bug Number	Description
3-16378591971	27379542	Customer Updater timed out when processing the productInfoChange, DiscountInfoChange and SponsorInfoChange events. This has been fixed.
NA	27380723	When the tax code was configured at the balance impact level, it was set to null in ECE. This has been fixed.
3-15392504701	27387822	Whenever Customer Loader failed, a high CPU utilization was observed. This has been fixed.
3-16661346291	27397772	Changing the credit limit for a specific resource was triggering unnecessary breach notifications. This has been fixed.
3-16054383111	27409225	When a credit-control request (CCR) was sent to Diameter Gateway for an account that did not exist in the cache, Diameter Gateway was returning an incorrect error code. This has been fixed.

Known Problems in ECE 11.3 Patch Set 6

This section provides an overview of the known problems in ECE 11.3 Patch Set 6.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)

ECE 11.3 Patch Set 5 Release Notes

This section provides information about ECE 11.3 Patch Set 5.

New Features in ECE 11.3 Patch Set 5

This section provides documentation for the features introduced in ECE 11.3 Patch Set 5.

Support for Monitoring ECE

ECE now provides an application to monitor ECE. The new ECE Monitoring Agent monitors the main ECE components and generates periodical reports and alerts on the system's availability and performance metrics. You can use the ECE Monitoring Agent to monitor:

- **ECE cluster.** Monitors the state of each ECE node configured with JMX port in the ECE topology and generates alerts in case of node failures, shutdowns, or network failures. For ECE charging server nodes, the Monitoring Agent generates alerts when there are unbalanced partitions across ECE server nodes and when the number of running storage-enabled nodes falls below the threshold configured.
- **ECE caches.** Monitors the ECE cache based on the estimated size of the cache and the threshold configured and generates alerts when the threshold breach occurs. You can define the threshold for generating alerts; for example, you can configure

the threshold percentage to generate an alert when the threshold is approaching or when the threshold breach occurs.

You can configure to monitor the following caches:

- Subscriber-based caches, such as Customer and Balance caches, based on the estimated size and the number of nodes available.
- Session-based caches, such as ActiveSession and RatedEvent caches, based on the number of nodes configured and the size of each node.
- **WebLogic Server and Oracle NoSQL database connections.** Monitors the connections to WebLogic Server and the Oracle NoSQL database storage nodes and reports connection failures encountered while publishing notifications or storing rated events into the database.
- **ECE server response time.** Monitors the ECE server response time for the different types of ECE requests received during the monitoring cycle; for example, IUT requests. You can define the response time and configure the Monitoring Agent to generate alerts in case of threshold breaches. The information about the requests are included in the **DiameterGatewayStatistics** and **EMGatewayStatistics** sections in the summary report logs.
- **Rated event throughput.** Monitors the number of rated events stored or retrieved from the cache or from the Oracle NoSQL database and generates alerts on threshold breaches.

In addition to generating periodical reports and alerts on threshold breaches, the Monitoring Agent reports any unusual scenarios that prevents ECE from continuing the processes. By default, the Monitoring Agent generates the following types of alerts: Warning, Critical, and Fatal. You can also define the conditions for generating the alerts; for example, you can define to generate alerts only five times in an hour.

To configure monitoring of ECE, a sample file named **monitor-configuration.xml**, is included in the *ECE_home/config* directory. You can use this file to configure the settings for monitoring ECE. See "[Configuring Monitoring of ECE](#)" for more information.

The Monitoring Agent runs periodically based on the configured intervals. It also runs when certain event-based scenarios occur; for example, node failures.

Types of Log Files

The Monitoring Agent generates the following types of log files:

- [monitorAgentX.log](#)
- [monitorAgentX_ECE_SUMMARY_REPORT.log](#)
- [monitorAgentX_ECE_ALERT_REPORT.log](#)

These log files are stored in the *ECE_home/occeserver/logs* directory by default. Review these log files regularly to monitor your system and detect and diagnose system problems.

monitorAgentX.log

This log file contains general information about various activities of the Monitoring Agent. This log provides information about the issues encountered by the Monitoring Agent.

monitorAgentX_ECE_SUMMARY_REPORT.log

This log file contains the summary report generated at regular monitoring intervals in the JSON format. The last section in the log file contains the details and the format of the information provided in the summary logs. By default, the Monitoring Agent creates 24 summary log files in a day (one summary log file for each hour of the day). After 24 hours, it starts deleting the summary log files one by one every hour and adds the summary log files for the current day. You can view a maximum of 24 summary log files at any time of a day.

The following is an example of the summary log generated in an hour:

```
{
  "SummaryReport" : {
    "ReportHeader" : {
      "NumberOfMembers" : 12,
      "ClusterSize" : 12,
      "ClusterName" : "BRM",
      "Availability" : "Running",
      "Time" : "2017-11-30 19:58:13",
      "State" : "UsageProcessing"
    },
    "ClusterMembers" : [ {
      "Role" : "server",
      "HeapSize" : 1073741824,
      "Availability" : "Running",
      "MemberName" : "ecs1",
      "Machine" : "url"
    }, {
      "Role" : "customerUpdater",
      "HeapSize" : 1073741824,
      "Availability" : "Running",
      "MemberName" : "customerUpdater1",
      "Machine" : "url"
    }, {
      "Role" : "pricingUpdater",
      "HeapSize" : 1073741824,
      "Availability" : "Running",
      "MemberName" : "pricingUpdater",
      "Machine" : "url"
    }, {
      "Role" : "brmGateway",
      "HeapSize" : 1073741824,
      "Availability" : "Running",
      "MemberName" : "brmGateway",
      "Machine" : "url"
    }, {
      "Role" : "emGateway",
      "HeapSize" : 1073741824,
      "Availability" : "Running",
      "MemberName" : "emGateway1",
      "Machine" : "url"
    }, {
      "Role" : "ratedEventFormatter",
      "HeapSize" : 1073741824,
      "Availability" : "Running",
      "MemberName" : "ratedEventFormatter1",
      "Machine" : "url"
    }, {
      "Role" : "diameterGateway",
      "HeapSize" : 1073741824,
```

```
    "Availability" : "Running",
    "MemberName" : "diameterGateway1",
    "Machine" : "url"
  } ],
  "SubscriberCacheUtilization" : [ {
    "UtilizationPercent" : 23,
    "Size" : 762600,
    "CacheName" : "Customer",
    "NumberOfEntries" : 659
  }, {
    "UtilizationPercent" : 21,
    "Size" : 190360,
    "CacheName" : "Balance",
    "NumberOfEntries" : 659
  } ],
  "SessionCacheUtilization" : [ {
    "MemberName" : "ecs1",
    "UtilizationPercent" : 0,
    "Size" : 0,
    "CacheName" : "ActiveSession",
    "NumberOfEntries" : 0
  }, {
    "MemberName" : "ecs1",
    "UtilizationPercent" : 0,
    "Size" : 0,
    "CacheName" : "RatedEvent",
    "NumberOfEntries" : 0
  }, {
    "MemberName" : "ecs1",
    "UtilizationPercent" : 32,
    "Size" : 5136,
    "CacheName" : "ServiceContext",
    "NumberOfEntries" : 12
  }, {
    "MemberName" : "ecs1",
    "UtilizationPercent" : 0,
    "Size" : 0,
    "CacheName" : "RecurringBundleIdHistory",
    "NumberOfEntries" : 0
  } ],
  "DiameterGatewayStatistics" : {
    "IUT" : {
      "MinLatency" : -9223372036854,
      "AverageLatency" : 0,
      "Throughput" : 0,
      "ThresholdLatency" : 48,
      "Count" : 0,
      "LatencyPercentile" : 0,
      "MaxLatency" : 9223372036854
    },
    "PerRequestType" : {
      "ExternalTopUpUpdate" : {
        "MinLatency" : 9,
        "AverageLatency" : 11,
        "Throughput" : 0.133,
        "ThresholdLatency" : 48,
        "Count" : 4,
        "LatencyPercentile" : 100.000,
        "MaxLatency" : 12
      }
    }
  }
}
```

```

    },
    "AllRequest" : {
      "MinLatency" : 9,
      "AverageLatency" : 11,
      "Throughput" : 0.133,
      "ThresholdLatency" : 48,
      "Count" : 4,
      "LatencyPercentile" : 100.000,
      "MaxLatency" : 12
    }
  },
  "EMGatewayStatistics" : {
    "AllRequest" : {
      "MinLatency" : 9223372036854,
      "AverageLatency" : 0,
      "Throughput" : 0,
      "ThresholdLatency" : 20,
      "Count" : 0,
      "LatencyPercentile" : 0,
      "MaxLatency" : -9223372036854
    }
  },
  "RatedEventThroughput" : {
    "RatedEventProcessor" : {
      "RatingThroughput" : 0,
      "ProcessingThroughput" : 0,
      "Ratio" : 0
    },
    "RatedEventFormatter" : {
      "PublishingThroughput" : 0,
      "ProcessingThroughput" : 0,
      "Ratio" : 0
    }
  }
}

```

monitorAgentX_ECE_ALERT_REPORT.log

This log file contains the alerts logged at regular monitoring intervals in the JSON format. The Alert Format section in this file specifies the formats and the types of alerts logged. The Monitoring Agent publishes the alerts as notifications to a JMS queue. By default, the Monitoring Agent creates 24 alert log files in a day (one alert log file for each hour of the day). After 24 hours, it starts deleting the alert log files one by one every hour and adds the alert log files for the current day. You can view a maximum of 24 alert log files at any time of a day.

You can also view these notifications and subscribe to receive the notifications by accessing the **ECE Monitoring.Notifier** node. See ["Subscribing Notifications"](#) for more information.

The following is an example of the alert log generated for the balance cache utilization:

```

{
  "Alert" : [ {
    "Name" : "BalanceCacheUtilization",
    "Type" : "Warning",
    "Source" : "ecs1"
  }, {
    "Target" : {
      "Percent" : 60
    }
  }
]
}

```

```

        "Basis" : "16000"
    }
}, {
    "Actual" : {
        "Percent" : 69
    }
} ]
}

```

Configuring Monitoring of ECE

To configure monitoring of ECE:

1. Open the `ECE_home/config/monitor-configuration.xml` file.
2. Specify or update the value for the entries listed in [Table 1-7](#) as appropriate.

Table 1-7 Entries in the Monitoring Configuration XML File

Entry	Description
<code>nodeHealthCheckInterval</code>	<p>The regular interval (in seconds) in which the Monitoring Agent is run.</p> <p>Note: The maximum and the default interval is 30 seconds.</p>
<code>alertCountResetInterval</code>	The interval (in seconds) for resetting the alert count.
<code>fatalAlertCount</code>	The number of fatal alerts generated after which the alerts are suppressed for the time defined by <code>alertCountResetInterval</code> .
<code>criticalAlertCount</code>	The number of critical alerts generated after which the alerts are suppressed for the time defined by <code>alertCountResetInterval</code> .
<code>warningAlertCount</code>	The number of warnings generated after which they are suppressed for the time defined by <code>alertCountResetInterval</code> .
<code>iutThresholdLatency</code>	<p>The latency (in milliseconds) for the Initiate, Update, and Terminate requests from Diameter Gateway to the ECE server.</p> <p>Note: Ensure that you enter the appropriate latency for generating this report; for example, a 99.99% or 100% latency.</p> <p>When <code>iutThresholdLatency</code> is set, the Monitoring Agent tracks the percentage of requests that are below the <code>iutThresholdLatency</code> value and generate the alerts based on <code>alertType</code> and <code>alertValue</code>.</p> <p>For example, if <code>iutThresholdLatency</code> is set to 10 and out of 100 calls made, 95 calls are less than or equal to 10 milliseconds, the latency percentile is considered as 95. If <code>alertType</code> is set to Warning and <code>alertValue</code> for Warning is set to 95, the Monitoring Agent checks if the latency percentile is less than <code>alertValue</code>, which is 95. In this case, the latency percentile is not less than <code>alertValue</code> for Warning, therefore the Monitoring Agent does not generate a Warning alert. In case if the latency percentile is 94, the Monitoring Agent generates a Warning alert.</p>

Table 1–7 (Cont.) Entries in the Monitoring Configuration XML File

Entry	Description
thresholdLatency	<p>The latency (in milliseconds) for all the traced requests from EM Gateway to the ECE server.</p> <p>When thresholdLatency is set, the Monitoring Agent tracks the percentage of requests that are below the thresholdLatency value and generates alert based on the alertType and alertValue.</p> <p>For example, if thresholdLatency is set to 10 and out of 100 requests received, 89 requests took less than or equal to 10 milliseconds, the latency percentile is considered as 89. If alertType is set to Critical and alertValue for Critical is set to 90, the Monitoring Agent checks if the latency percentile is less than alertValue, which is 90. In this case, the latency percentile is less than alertValue for Critical, therefore the Monitoring Agent generates a Critical alert. If the latency percentile is equal to or more than alertValue, the Monitoring Agent does not generate a Critical alert. However, if the latency percentile is less than the alertValue for Warning, the Monitoring Agent generates a Warning alert.</p>
alertType	<p>The type of alert generated. The following are the valid alert types: Warning, Critical, and Fatal.</p>

Table 1–7 (Cont.) Entries in the Monitoring Configuration XML File

Entry	Description
alertValue	<p>The value at which the alert is generated. You specify the value for this entry as follows:</p> <ul style="list-style-type: none"> ■ In the diameterGatewayLatency and emGatewayLatency sections, you specify the percentages for tracking requests from Diameter Gateway and EM Gateway. For example, you can configure 95% latency for Warnings, 90% latency for Critical alerts, and 80% latency for Fatal alerts. ■ In the ratedEventThroughput section, you specify the ratio for tracking rated events throughput. This is the ratio of the number of rated events stored or retrieved from the cache or from the Oracle NoSQL database. For example, you can configure 0.95 for Warnings, 0.90 for Critical alerts, and 0.80 for Fatal alerts. ■ In the partitionsUnbalanced section, you specify the number of occurrences for tracking unbalanced partitions across ECE server nodes. For example, you can configure 3 occurrences for Warnings, 10 occurrences for Critical alerts, and 20 occurrences for Fatal alerts. ■ In the runningServers section, you specify the number of server nodes that are currently running for tracking the server nodes availability. For example, you can configure 2 nodes for Warnings, 1 node for Critical alerts, and 0 node for Fatal alerts. ■ In the noSQLCommitFailure and webLogicPublishFailure sections, you specify the number of connection failures at which the alerts are generated. For example, you can configure 1 failure for Warnings, 3 failures for Critical alerts, and 5 failures for Fatal alerts. ■ In the subscriberCacheUtilization section, you specify the percentage for tracking the total subscriber cache size (in bytes) across all the ECE server nodes; for example, reaching 60% of cache size for Warnings, reaching 80% of cache size for Critical alerts, and reaching 100% of cache size for Fatal alerts. ■ In the sessionCacheUtilization section, you specify the percentage for tracking the total session cache size (in bytes) for a ECE server node; for example, reaching 60% of cache size for Warnings, reaching 80% of cache size for Critical alerts, and reaching 100% of cache size for Fatal alerts.
subscriberCacheCapacity name	The name of the subscriber-related caches, such as Customer and Balance caches.
projectedClusterCapacity	The projected capacity of all ECE server nodes to hold a configured subscriber-based cache, such as Customer and Balance caches.
sessionCacheCapacity name	The name of the session-related caches, such as ActiveSession, RatedEvent, ServiceContext, and RecurringBundleIdHistory.
projectedNodeCapacity	The projected capacity of a ECE server node to hold a configured session-based cache, such as the ActiveSession cache.

3. Save and close the file.
4. On the machine where you have Elastic Charging Controller (ECC) installed, go to `ECE_home/occeserver/bin`.
5. Start ECC:

`./ecc`

6. Run the following command, which deploys the ECE installation onto the server machines:

sync

The **sync** command copies the relevant files of the ECE installation onto the server machines in the ECE cluster.

7. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - b. Start the ECE Monitoring Agent nodes (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 Monitoring Agent 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 Monitoring** node.
8. Expand a **MonitorConfiguration** node; for example, **MonitorConfiguration.diameterGatewayLatency**.
9. Expand **Attributes**.
10. Verify that the values that you specified in step 2 appears.

Note: The attributes displayed here are *read-only*. You can update these attributes by editing the `ECE_home/config/monitor-configuration.xml` file.

Subscribing Notifications

To subscribe for receiving monitoring notifications:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - b. Start the ECE Monitoring Agent nodes (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 Monitoring Agent 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 Monitoring** node.
2. Expand the **Notifier** node.
3. Expand **Attributes**.
4. Click **Notifications**.

The monitoring notifications published to the JMX notification queue appears.

5. Click **Subscribe**.

Pricing and Configuration Data Can Now Be Persisted in Active Persistence Mode

Important: This feature is supported only in the test environment.

PDC publishes the pricing and configuration data to ECE. If there are large volumes of data, the time taken to restart the ECE server and publish the data to ECE was slightly more.

With this enhancement, the pricing and configuration data is persisted in the active persistence mode when the first time the data is published to ECE and this data is used for processing requests during the subsequent restarts of the ECE server. You can enable active persistence by setting the **java.property.ece.persistence.mode** entry in the *ECE_home/occeserver/config/ece.properties* file. See "[Enabling Active Persistence](#)" for more information.

About Active Persistence of ECE Caches

When the active persistence mode is enabled, ECE enables active persistence of the following cache schemes: `ReplicatedFederatedCache` and `XRefFederatedCache`. ECE persists all the ECE caches that use these schemes each time these caches are updated. ECE does not persist the customer data stored in `XRefFederatedCache` and also the ECE state.

About Storing Persisted Data

By default, the persisted data files are stored in the *ECE_home/persistence* directory. You can change this location by setting the **java.property.coherence.distributed.persistence.base.dir** entry in the *ECE_home/occeserver/config/ece.properties* file. See "[Changing Persistence Location](#)" for more information.

In a multi-node configuration, you can use a network file system so that all the data persisted across multiple server nodes can be stored in the same location. In case if you are using a local file system to store the persisted data files, the persisted data from the respective servers local file system are loaded into the cache when you restart the server.

Enabling Active Persistence

To enable the active persistence:

Note: Enabling or disabling the active persistence requires complete restart of the ECE system and cannot be performed on a running ECE system.

1. On the driver machine, open the *ECE_home/occeserver/config/ece.properties* file.
2. Search for the following entry:

```
java.property.ece.persistence.mode=on-demand
```

3. Set the entry to **active** to enable active persistence:

```
java.property.ece.persistence.mode=active
```

4. (Optional) Search for the following entry and enter the path to the directory in which you want to store the persisted data files:

```
java.property.coherence.distributed.persistence.base.dir=../persistence/
```

5. Save and close the file.
6. On the machine where you have ECC installed, go to *ECE_home/occeserver/bin*.

7. Start ECC:

```
./ecc
```

8. Run the following command, which deploys the ECE installation onto the server machines:

```
sync
```

The **sync** command copies the relevant files of the ECE installation onto the server machines in the ECE cluster.

When you restart the ECE server, the pricing and configuration data is persisted in the active persistence mode.

Changing Persistence Location

To change the persistence location:

1. On the driver machine, open the *ECE_home/occeserver/config/ece.properties* file.
2. Search for the following entry:

```
java.property.coherence.distributed.persistence.base.dir=ECE_home /persistence
```

3. Set the path to the new persistence data directory:

```
java.property.coherence.distributed.persistence.base.dir=persistence_base_dir
```

where *persistence_base_dir* is the path to the directory in which you want to store the persisted data files.

4. Save and close the file.
5. On the machine where you have ECC installed, go to *ECE_home/occeserver/bin*.
6. Start ECC:

```
./ecc
```

7. Run the following command, which deploys the ECE installation onto the server machines:

```
sync
```

The **sync** command copies the relevant files of the ECE installation onto the server machines in the ECE cluster.

ECE Enhancement to Process Parallel Requests for the Same Customer

In the previous releases, ECE did not consider parallel requests received for the same customer for processing. With this enhancement, ECE processes the usage requests equally for all the accounts by queuing the requests until the present charging cycle is complete.

ECE Now Supports Match Factor in Discount Calculation

In previous releases, the match factor in discounting was supported only in BRM. In BRM, the match factor is used with cascading discounts in which a discount can be applied only to the remaining portion of usage that is not discounted before.

However, in ECE, when a discount was based on quantity, the whole quantity was considered for the discount instead of only the remaining portion.

ECE now supports match factor in discounting. You can enable match factor in ECE by setting the **isMatchFactorEnabled** entry in the *ECE_*
home/occeserver/config/management/charging-settings.xml file to **true**.

To enable match factor in ECE:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand **charging.server**.
3. Expand **Attributes**.
4. Set the **matchFactorEnabled** attribute to **true**.

High Availability Support for ECE Driver Machine

In ECE, high availability of the driver machine is required primarily to synchronize the configuration across all the machines running ECE nodes. In previous releases, when the driver machine was down, the ECE configuration could not be synchronized across all the machines.

You can now configure a primary driver machine and a backup driver machine that takes over when the primary machine goes down or fails. You can configure the backup driver machine by setting the **backupDriverIP** entry in the *ECE_*
home/occeserver/config/ece.properties file. When the backup driver machine is configured:

- The ECE configuration in both the driver machine and the backup driver machine is updated each time an ECE MBean is set.
- If the primary or backup driver machine goes down or if it is not reachable, the ECE configuration in the driver machines is not updated when an ECE MBean is set.

To configure a new or backup driver machine:

If you have a driver machine and multiple server machines in your system, you can configure one of the existing server machines to be the new driver machine or the backup driver machine. Before you configure the driver machine:

- Ensure that the selected machine has a JMX-management-enabled server node running.

- Ensure that the two-way password-less SSH connection is established between the selected machine and all the other machines in your topology.
 - Ensure that the primary driver machine is added to the ECE topology. This ensures that the primary driver machine is also considered for synchronization when the `sync` command is run.
1. On the selected machine, do one of the following:
 - To configure the selected machine as the new driver machine, run the following command:


```
setDriverMachine -name driverIP -ip Driver_IP_Address
```

where *Driver_IP_Address* is the IP address of the machine selected to be the new driver machine.

Running this command without the `-name driverIP` parameter configures the selected machine as the primary driver machine.
 - To configure the selected machine as the backup driver machine, run the following command:


```
setDriverMachine -name backupDriverIP -ip IP_Address
```

where *Backup_IP_Address* is the IP address of the machine selected to be the backup driver machine.

The selected machine is assigned as the new or backup driver machine and a success message appears.
 2. Verify that the new or backup driver machine is performing the relevant operations.

Additional Attributes for Reservation Quota Management

In previous releases, you could not configure validity, quota holding time, and quota threshold based on the attributes for reservation quotas granted as part of authorization or reauthorization.

In ECE, you can now configure the way dynamic quotas are managed by setting the following quota attributes to optimize network signaling traffic:

- **Quota holding time (QHT).** Specifies how long a granted quota can be idle before the reservation is released.
- **Volume quota threshold (VQT).** Specifies how much of the granted quota must be consumed before a subscriber can request additional quota. This attribute is configured per service, event, and number of granted units.
- **Validity time (VT).** Specifies whether VT can be set to a fixed value per service-event combination at runtime. This attribute is independent of the number of units in the granted quota.

Diameter Gateway now populates QHT, VQT, and VT attribute-value pairs (AVPs) in the Credit-Control-Answer message if the corresponding attributes are present in the usage response received from the ECE server.

You can use the pre-rating extension to override the QHT, VQT, and VT values in the usage request. For more information on using the pre-rating extension, see *ECE Extensions*. See the **SampleDynamicQuotaExtension** sample program in the ECE SDK (`ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions`

directory) for an example of how to implement a custom logic to override the QHT, VQT, and VT values by using the pre-rating extension.

You can also use the post-charging extension to override the QHT and VQT values in the usage response. For more information on using the post-charging extension, see *ECE Extensions*. See the **SamplePostChargingExtension** sample program in the ECE SDK for an example of how to implement a custom logic to override the QHT and VQT values by using the post-charging extension.

Pre-Rating Extension Enhanced to Support Dynamic Quota

When reserving resources for a session request, ECE bases the reservation amount on the requested service units (RSUs) in the request. To support dynamic quotas, you must be able to modify the number of RSUs in a request to address the needs of concurrent sessions. To enable this, the pre-rating extension has been enhanced to provide access to additional customer, product, and balance data.

You can use this data to derive a quota if the existing configuration requires modification to suit your business requirements. You can then use the derived quota to update the usage request by using the pre-rating extension. For information on using the pre-rating extension, see *ECE Extensions*.

See the **SamplePreRatingExtension** sample program in the ECE SDK (*ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions* directory) for an example of how to implement a custom logic to modify the number of RSU in the usage request by using the pre-rating extension.

Post-Charging Extension Enhanced to Support URL Redirection

In previous releases, you could redirect subscriber sessions only by using the redirection configuration in ECE. ECE now allows you to add or modify redirection rules by using the post-charging extension. For more information on using the post-charging extension, see *ECE Extensions*.

For more information on redirection-rule conditions for adding or modifying redirection rules, see the discussion about redirecting a subscriber session to a service portal in *ECE Implementation Guide*.

See the **SamplePostChargingExtension** sample program in the ECE SDK (*ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions* directory) for an example of how to implement a custom logic to add or modify redirection rules by using the post-charging extension.

Service Transfer Data Updates Are Now Synchronized in Real Time

In BRM, when a service is transferred from one account to another, objects associated with the service and member services are transferred to the new account. This data is published to ECE so that the usage for the transferred service can be allocated to the new account appropriately.

In previous releases, such updates were not published to ECE and some customization was required in BRM to publish these updates to ECE.

With this enhancement, service transfer updates are published to ECE synchronously. This maintains the synchronization of the service transfer data updates in BRM and ECE and ensures that the usage is allocated to the appropriate account.

Support for GC Logs

In previous releases, the Garbage Collection (GC) debug logs were not supported.

With this enhancement, for ECE processes started by running the **ecc** command, the GC debug logs are enabled by default. The GC debug log files are stored in the ECE logs directory (*ECE_home/occeserver/logs*).

If you suspect a problem with these processes, you can look in the *ECE_home/occeserver/logs/Instance_Name_GC.log* files for errors, where *Instance_Name* is the name of the ECE process-node instance (the name of the process you defined in the ECE topology file) that you need to troubleshoot. For example, look in **emGateway1_GC.log**. By default, four GC log files are made available in the directory, you can change the number of GC log files available by setting the **numberOfGCLogFiles** in the *ECE_home/occeserver/config/ece.properties* file.

You can also disable the generation of GC log files by setting the **enableGCLogs** entry in the *ECE_home/occeserver/config/ece.properties* file.

Collecting GC Log Files

You can now use the **infoCollector** command to collect GC logs.

To collect GC log files, run the following **infoCollector** command with the **-gc** parameter:

```
infoCollector [-v] [-nc] [-l] [-gc] [-d dir][-t]-td dir [-s] [-e "FileFilter",
"FileFilter2", "..."]
```

where:

- **-v** turns on verbose mode that outputs information pertaining to the collected files; this option provides status of what is being copied as the command collects files.
- **-nc** does *not* compress the resulting directory into a compressed TAR file.
- **-l** includes all log files into the collection; if **-l** is omitted, the command collects only those log files that match the node-name with the **.log** suffix that you specify.
- **-gc** collects all the GC log files.
- **-d dir** uses the directory you specify to hold all collected data where *dir* is the path of the directory.
- **-t** adds all the trace files matching the subscription session identifiers to the collection.
- **-td dir** collects all the trace files from the provided directory or location.
- **-s** includes all files from the *ECE_home/occeserver/sample_data* directory.
- **-e "FileFilter"** is the path name or path name pattern of custom directories or files to collect and include in the compressed TAR file. Separate multiple filters with a comma.

Summary of Fixes in ECE 11.3 Patch Set 5

[Table 1–8](#) lists the bugs that were fixed in ECE 11.3 Patch Set 5 and provides a brief description of the resolution.

Table 1–8 Bug Fixes in ECE 11.3 Patch Set 5

SR Number	Bug Number	Description
3-14737482431	26286323	The threshold percentages in the threshold breach notifications were not in the correct order. This has been fixed.
3-14793395681	26301135	The Public User Identity information was blank while processing the threshold breach event notification. This has been fixed.
3-15119738151	26313107	The rating in ECE with RUM-based evaluation was producing incorrect balance impacts. This has been fixed.
3-15005317451	26412350	When using the BypassOCS method, the Diameter packet was causing an exception which resulted in incorrect CCA. This has been fixed.
3-15289774011	26448291	When processing large number of usage requests, the ECE process was getting into a loop without sending the responses. This has been fixed.
3-15223373621	26520202	ECE was not supporting system-level charge sharing. This has been fixed.
3-15283213601	26544011	In a session, the fixed alteration packets were getting considered for split-charge periods and was causing incorrect split charging. This has been fixed.
3-15301908510	26545432	When alterations were applied to a multi-RUM impact, there was an error with the functor evaluation. This has been fixed.
3-15315605011	26567936	When implementing custom AVPs and the logic to work with those custom AVPs through extension hooks, casting error was encountered for such AVPs. This has been fixed.
NA	26568087	ECE was required to execute the preprocessor applycontextchanges method changes only if BypassOCS was set to true . This has been fixed.
3-15301908491	26582157	ECE was discounting more than 100% in some cases. This has been fixed.
3-15370240271	26608630	When Diameter Gateway was queried for balances for a non-existent account, it was returning an incorrect error code, DIAMETER_UNABLE_TO_COMPLY in CCA. This has been fixed.
3-15170773001	26658030	ECE required balance extension to query balances based on the requested timestamp. This has been fixed.

Table 1–8 (Cont.) Bug Fixes in ECE 11.3 Patch Set 5

SR Number	Bug Number	Description
3-15591313781	26704797	The profile label information was not updated from BRM to the ECE cache, which is causing incorrect charging due to missing label. This has been fixed.
3-15593196271	26704802	During usage rating, ECE was generating notification publish errors in the ECS log files. This has been fixed.
3-1551947734	26799046	The discounting was incorrectly done due to wrong calculation of total quantity information. This has been fixed.
3-15580003091	26799056	ECE was not generating breach notification even for 100% usage. This has been fixed.
3-15456864501	26799058	The profile objects deleted from BRM were not reflected in the ECE cache causing incorrect rating results. This has been fixed.
3-15688688171	26799068	The pricing updater log messages were at INFO level when run in the DEBUG mode. This has been fixed.
3-15696534961	26807268	When a usage request was rejected for billing due, ECE was throwing a null pointer exception. This has been fixed.
3-15776018131	26849725	When ECE could not map a specific combination of service and event to a specific product, it was throwing the "PRODUCT_NOT_FOUND" error and subsequently Diameter Gateway was returning an incorrect error, DIAMETER_USER_UNKNOWN. This has been fixed.
3-15645194261	26944337	ECE was throwing Out of Memory exception when the rate plan was large. This has been fixed.
3-15182173641	27007662	The rating was failing for delayed usages after the service update followed by change of plan. This has been fixed.
3-15855361491	27044921	During usage rating if the used volume was measured in KB, then ECE was rating the usage incorrectly. This has been fixed.
3-16080672781	27108850	ECC was not working as expected when coherence parameters were passed after tuning parameters into JavaCommandLineBuilder. This has been fixed.
3-15747288711	27108882	The usage was charged incorrectly when there was a plan change for a closed-user-group member without the profile sharing group. This has been fixed.

Table 1–8 (Cont.) Bug Fixes in ECE 11.3 Patch Set 5

SR Number	Bug Number	Description
3-14784429761	27109002	The customer details were not made available in the ECE cache when a new account was created in BRM even if the balance information was correct and the usage for that account was rated properly. This has been fixed.
3-15614113231	27109006	Customer Updater process was not launching. This has been fixed.
3-14678315081	26313176 26259420	In a ECE system with PCRF, once after diameter-peer-request and diameter-peer-answer was exchanged between Diameter Gateway and PCRF, ECE was not publishing the SNR notification. This has been fixed.

Known Problems in ECE 11.3 Patch Set 5

This section provides an overview of the known problems in ECE 11.3 Patch Set 5.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)

ECE 11.3 Patch Set 4 Release Notes

This section provides information about ECE 11.3 Patch Set 4.

New Features in ECE 11.3 Patch Set 4

This section provides an overview of the features introduced in ECE 11.3 Patch Set 4.

Diameter Gateway Enhancement for Policy Counters

For Sy subscriptions, you can now configure ECE to reject a Spending Limit Request (SLR) if there are no policy counters available for the subscriber. You can do this by setting the **syRejectNoCounters** entry in the `ECE_home\ooceserver/config/management/charging-settings.xml` file to **true**.

To configure Diameter Gateway to reject SLRs when no policy counters are available:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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\ooceserver/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.policyConfig**.
3. Expand **Attributes**.

4. Set the `syRejectNoCounters` attribute to `true`.

ECE Now Supports Incremental Rating for Mid-session Rating Condition Changes

You can now configure ECE to apply mid-session rating condition changes to data sessions that generate a reauthorization request (RAR) and affect the session's rating. When changes in charging occur during an ongoing data session, they trigger a RAR. You can configure ECE to generate a rated event whenever charging changes are triggered during the session. In all scenarios, ECE considers changes in charging conditions for the portion of the session for which they are applicable.

Note: In this release, incremental rating for mid-session rating condition changes is supported only for those sessions that generate a RAR.

You can configure ECE to use incremental rating for mid-session rating condition changes by setting the `enabledOrDisableNonLinear` entry in the `ECE_home/occeserver/config/management/charging-settings.xml` file to `true`.

To configure ECE to use incremental rating for mid-session rating condition changes:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand `charging.reservationConfig`.
3. Expand **Operations**.
4. For each product that you offer, do the following:
 - a. Select `enabledOrDisableNonLinear`.
 - b. Specify values for the following parameters:
 - productType.** Enter the name of the product defined in the ECE request specification data (for example, DATA).
 - enableOrDisable.** Enter `true` to use incremental rating for mid-session rating condition changes.
 - c. Click the `enabledOrDisableNonLinear` button.

The following sections describe a couple of scenarios of a non-predictable rating condition change during a data session and the balances and charges during and at the end of the session.

Scenario: Noncurrency Voucher Top-up During a Session

Given a subscriber with:

- A charge offer named *CO Usage Data*
- A charge of \$1 per 1 megabyte (MB) of usage
- A discount offer which consumes from available free MB
- An initial balance of 0 bytes of *Free Use Data* and USD = 0
- Incremental rating enabled for product type *TelcoGPRS*
- Eligible offers selected based on duration

The usage charging flow is as follows:

1. The subscriber starts a data session at 23:50 PM on April 1 with an INITIATE request for 30 MB.
2. Granted 40 MB with validity of 1 hour.
3. Reserves the balance USD 30.
4. When non-currency voucher top-up is done at 00:00 hours, which grants 10 MB of data with validity of 5 days, a RAR notification is sent.
5. The UPDATE request is received with requested units as 40 MB and used units as 12 MB at 00:02 on April 3.
6. A CDR is generated for 12 MB (23:50:00-00:00:00 April2 (10MB @1/MB) =10\$, 00:00:00-00:02:00 (10MB -2Mb) = -8MB).
7. Granted 40 MB with validity of 1 hour.

Balances will be:

- Current balance: USD 10
 - Free MB =-8MB
8. The TERMINATE request is received with Used units as 20 MB at 01:30 on April 3.

Balances will be:

- Current balance: USD 22
- Free MB = 0

Scenario: Recharge Consumed in a Session

Given a subscriber with a charge offer named *Data Standard*, with a *Standard Bytes* (chargeable balance) followed by *Free Use Data* (counter), with a starting balance of 50 MB *Standard Bytes* and 0 bytes of *Free Use Data*.

Given a subscriber with:

- A charge offer named *Data Standard*
- A consumption cascade of *Standard Bytes* (chargeable balance) followed by *Free Use Data* (counter)
- A starting balance of 50 MB *Standard Bytes*
- 0 bytes of *Free Use Data*

The usage charging flow is as follows:

1. The subscriber starts a data session at 1:00 PM
2. Consumes 50 MB over 20 minutes at high quality of service (QoS)
3. Consumes 2 MB over 5 minutes at low QoS

4. At 1:25 PM, recharges with 100 MB *Standard Bytes* valid for 30 minutes
5. Consumes 100 MB over 20 minutes at high QoS
6. Consumes 5 MB over 15 minutes at low QoS before terminating the session

Table 1–9 shows the subscriber’s balances and charges at the end of the session.

Table 1–9 Balances and Charges

Balance Name	Balance Value	Units Charged
Data Standard	0 MB	150 MB
Free Use Data	7 MB	7 MB

ECE Now Supports Subscriber-Based Tracing and Logging

In previous releases, if a problem was reported and debug was required, it was not possible to debug for the selective subscriber. It was difficult to trace the details of a particular subscriber from the logs. Also, the size of the log files had significant performance impact on the system.

ECE now provides the capability to selectively trace a subscriber’s session. You can selectively trace a particular subscriber’s session based on the subscriber ID. You can now specify a list of subscriber IDs to be traced. ECE then generates log files for the listed subscribers for each session. If a particular subscriber has multiple sessions, separate log files are generated for each session.

The trace file names are unique and are in the format of *node_name.subscriber_ID.session_ID.log* file. For example, **ecs1.SUBSCRIBER1.SESSION1.log**.

Note: By default, ECE can trace and log a maximum of 100 subscriber IDs and 24 session IDs per subscriber at a time. However, if required, you can configure these parameters.

ECE does not archive or remove the log files that are generated. Make sure that you remove or archive the log files, periodically, to avoid running out of disk space.

To enable subscriber-based tracing, a new file named **subscriber-trace.xml**, is included in the *ECE_home/config* directory. You can update this file to add a list of subscriber IDs whose sessions need to be traced. See "[Enabling or Updating Subscriber-Based Tracing and Logging](#)" for more information.

You can also specify if you want to trace and log selective functions, such as alterations (discounts), charges, and distributions (charge sharing), for each subscriber specified in the list. To enable subscriber-based tracing and logging for selective functions, see "[Enabling or Updating Subscriber-Based Tracing and Logging for Selective Functions](#)".

By default, the session trace log files are stored in the *ECE_home/occeserver/logs* directory. However, you can modify the location of the session trace files by updating the **subscriberTraceLogDir** parameter in the *ECE_home/occeserver/config/ece.properties* file.

Note: By default, the log level is set to **DEBUG** (recommended log level) to generate logs with detailed information. However, you can change the log levels by updating the **logLevel** parameter in the *ECE_home/config/subscriber-trace.xml* file.

Enabling or Updating Subscriber-Based Tracing and Logging

To enable or update subscriber-based tracing and logging:

1. Open the *ECE_home/config/subscriber-trace.xml* file.
2. Search for the **subscriberTraceConfig** section.
3. Update the values for the following attributes if required:
 - **logMaxSubscribers**. Specify the maximum number of subscribers for whom you want to enable tracing. The default value is 100.
 - **logMaxSubscriberSessions**. Specify the maximum number of sessions for which the logs need to be generated per subscriber. The default value is 24.
 - **logLevel**. Specify the log level you want to use for generating logs; for example, DEBUG or ERROR. The default value is DEBUG.
4. In the **subscriberList** section, provide the list of subscriber IDs:

The following is an example of the XML file that includes the subscriberList section:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<config>
  <!--
    <subscriberTraceConfig
      logMaxSubscribers = "100"
      logMaxSubscriberSessions = "24"
      logExpiryWaitTime = "1"
      logCleanupInterval = "2"
      logLevel = "DEBUG"
    config-class="oracle.communication.brm.charging.subscribertrace.configuration.i
nternal.SubscriberTraceConfig">
      <subscriberList config-class="java.util.ArrayList">
        <subscriber id="6500000000">
          config-class="oracle.communication.brm.charging.subscribertrace.configuration.i
nternal.SubscriberImpl"/>
          <subscriber id="6500000001">
            config-class="oracle.communication.brm.charging.subscribertrace.configuration.i
nternal.SubscriberImpl"/>
        </subscriberList>
      </subscriberTraceConfig>
    </config>
```

5. Save and close the file.
6. On the machine where you have ECC installed, go to *ECE_home/occeserver/bin*.
7. Start ECC:

```
./ecc
```

8. Run the following command, which deploys the ECE installation onto the server machines:

```
sync
```

The **sync** command copies the relevant files of the ECE installation onto the server machines in the ECE cluster.

9. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.

- 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 Logging** node.
10. Expand **Configuration**.
 11. Expand **Operations**.
 12. Select **updateSubscriberTraceConfiguration**.
 13. Click the **updateSubscriberTraceConfiguration** button.
 14. In the editor's MBean hierarchy, expand the **ECE Subscriber Tracing** node.
 15. Expand **SubscriberTraceManager**.
 16. Expand **Attributes**.
 17. Verify that the values that you specified in step 3 appears.

Note: The attributes displayed here are *read-only*. You can update these attributes by editing the *ECE_home/config/subscriber-trace.xml* file.

Disabling Subscriber-Based Tracing and Logging

To disable subscriber-based tracing and logging:

1. Open the *ECE_home/config/subscriber-trace.xml* file.
2. Search for the **subscriberTraceConfig** section.
3. Remove all the subscriber IDs from the **subscriberList** section. For example:

```
<subscriber id="6500000001"
config-class="oracle.communication.brm.charging.subscribertrace.configuration.i
nternal.SubscriberImpl"/>
```

4. Save and close the file.
5. On the machine on which you have ECC installed, go to *ECE_home/occeserver/bin*.
6. Start ECC:

```
./ecc
```
7. Run the following command, which deploys the ECE installation onto the server machines:

```
sync
```

The **sync** command copies the relevant files of the ECE installation onto the server machines in the ECE cluster.

8. Access the ECE MBeans:

- a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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 Logging** node.
9. Expand **Configuration**.
 10. Expand **Operations**.
 11. Select **updateSubscriberTraceConfiguration**.
 12. Click the **updateSubscriberTraceConfiguration** button.

Enabling or Updating Subscriber-Based Tracing and Logging for Selective Functions

Important: You can enable subscriber-based tracing and logging only for the following selective functions: alterations, charges, and distributions.

To enable or update subscriber-based tracing and logging for selective functions:

1. Open the *ECE_home/config/subscriber-trace.xml* file.
2. Go to the **subscriberTraceConfig** section.
3. Update the values for the following attributes, if required:
 - **logMaxSubscribers**. Specify the maximum number of subscribers for whom you want to enable tracing. The default value is 100.
 - **logMaxSubscriberSessions**. Specify the maximum number of sessions for which the logs to be generated per subscriber. The default value is 24.
 - **logLevel**. Specify the log level you want to use for generating logs; for example, DEBUG or ERROR. The default value is DEBUG.
4. In the **subscriberList** section, provide the list of subscriber IDs:

The following is an example of the XML file that includes the **subscriberList** section:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<config>
  <!--
    <subscriberTraceConfig
      logMaxSubscribers = "100"
      logMaxSubscriberSessions = "24"
      logExpiryWaitTime = "1"
      logCleanupInterval = "2"
      logLevel = "DEBUG"
    config-class="oracle.communication.brm.charging.subscribertrace.configuration.i
nternal.SubscriberTraceConfig">
      <subscriberList config-class="java.util.ArrayList">
```

```

        <subscriber id="811000000000"
config-class="oracle.communication.brm.charging.subscribertrace.configuration.i
nternal.SubscriberImpl"/>
    </subscriberList>
</subscriberTraceConfig>
</config>

```

5. Search for the **ComponentLoggerList** section.

6. Do the following:

- a. To enable subscriber-based tracing and logging for the alteration function, uncomment the following **componentLogger** blocks in the **componentLoggerList** section:

```

<componentLoggerList config-class="java.util.ArrayList">
    <!-- Add specific logger name and log level to overwrite the
out of the box default logger level
        For example, the following configuration overwrite the
logger named
"oracle.communication.brm.charging.subscribertrace.configuration"
to have log level "ALL" and the logger named
"oracle.communication.brm.charging.ratedevent.publisher to have"
log level "OFF".
        The logger level can only be "ALL", "DEBUG", "ERROR",
"FATAL", "INFO", "OFF", "TRACE", "WARN"
    -->
    <componentLogger
        loggerName="ALL"
        loggerLevel="ERROR"
        config-class="oracle.communication.brm.charging.subscr
ibertrace.configuration.internal.ComponentLoggerImpl"/>
    <componentLogger
        loggerName="oracle.communication.brm.charging.rating.alt
eration"
        loggerLevel="DEBUG"
        config-class="oracle.communication.brm.charging.subscrib
ertrace.configuration.internal.ComponentLoggerImpl"/>
</componentLoggerList>

```

- b. To enable subscriber-based tracing and logging for the charging function, uncomment the following **componentLogger** blocks in the **componentLoggerList** section:

```

<componentLoggerList config-class="java.util.ArrayList">
    <!-- Add specific logger name and log level to overwrite the
out of the box default logger level
        For example, the following configuration overwrite the
logger named
"oracle.communication.brm.charging.subscribertrace.configuration"
to have log level "ALL" and the logger named
"oracle.communication.brm.charging.ratedevent.publisher to have"
log level "OFF".
        The logger level can only be "ALL", "DEBUG", "ERROR",
"FATAL", "INFO", "OFF", "TRACE", "WARN"
    -->
    <componentLogger
        loggerName="ALL"
        loggerLevel="ERROR"
        config-class="oracle.communication.brm.charging.subscr

```

```

ibertrace.configuration.internal.ComponentLoggerImpl"/>
    <componentLogger
        loggerName="oracle.communication.brm.charging.rating.c
harge"
        loggerLevel="DEBUG"
        config-class="oracle.communication.brm.charging.subscr
ibertrace.configuration.internal.ComponentLoggerImpl"/>
</componentLoggerList>

```

- c. To enable subscriber-based tracing and logging for the distribution (charge sharing) function, uncomment the following `componentLogger` blocks in the `componentLoggerList` section:

```

<componentLoggerList config-class="java.util.ArrayList">
    <!-- Add specific logger name and log level to overwrite the
out of the box default logger level
        For example, the following configuration overwrite the
logger named
"oracle.communication.brm.charging.subscribertrace.configuration"
to have log level "ALL" and the logger named
"oracle.communication.brm.charging.ratedevent.publisher" to have"
log level "OFF".
        The logger level can only be "ALL", "DEBUG", "ERROR",
"FATAL", "INFO", "OFF", "TRACE", "WARN"
-->
    <componentLogger
        loggerName="ALL"
        loggerLevel="ERROR"
        config-class="oracle.communication.brm.charging.subscr
ibertrace.configuration.internal.ComponentLoggerImpl"/>
    <componentLogger
        loggerName="oracle.communication.brm.charging.rating.di
stribution"
        loggerLevel="DEBUG"
        config-class="oracle.communication.brm.charging.subscri
bertrace.configuration.internal.ComponentLoggerImpl"/>
</componentLoggerList>

```

7. Save and close the file.
8. On the machine where you have ECC installed, go to `ECE_home/occeserver/bin`.
9. Start ECC:

```
./ecc
```

10. Run the following command, which deploys the ECE installation onto the server machines:

```
sync
```

The `sync` command copies the relevant files of the ECE installation onto the server machines in the ECE cluster.

11. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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 Logging** node.
12. Expand **Configuration**.
13. Expand **Operations**.
14. Select **updateSubscriberTraceConfiguration**.
15. Click the **updateSubscriberTraceConfiguration** button.
16. In the editor's MBean hierarchy, expand the **ECE Subscriber Tracing** node.
17. Expand **SubscriberTraceManager**.
18. Expand **Attributes**.
19. Verify that the values that you specified in step 3 appears.

Note: The attributes displayed here are READ ONLY. You can update these attributes by editing the *ECE_home/config/subscriber-trace.xml* file.

Collecting Subscriber-Based Log Files

You can now use the **infoCollector** command to collect subscriber-based tracing log files from your ECE system.

To collect subscriber-based log files, run the **infocollector** command with the following new parameters:

```
infoCollector [ [-td dir] [-t "SUBSCRIPTION|SUBSCRIPTION.SESSION IDENTIFIER", "..."]
```

where:

- **-td dir** collects all the trace files from the provided directory or location. For example:

```
infoCollector -td ECE_home/trace01
```
- **-t "SUBSCRIPTION|SUBSCRIPTION.SESSION IDENTIFIER", "..."** adds all the trace files matching the subscription session identifiers to the collection. For example, see the following:

```
infoCollector -t "0048100700.Session01", "0048100702"
```

The **infoCollector** command does not collect files from systems on which ECC is running. For example, the **infoCollector** command does not collect files from the network mediation system.

Log4j Changes for Configuring Logging

In previous releases, you used the *ECE_home/occeserver/config/log4j.properties* file to configure logging for the entire cluster.

The **Log4j.properties** file is now replaced with the **log4j2.xml** file. You can now use the **log4j2.xml** file in the *ECE_home/occeserver/config* directory to configure logging in the XML format for the entire cluster. Make sure that you edit this file before starting

the ECE charging servers. After starting the ECE charging servers, use JConsole to configure logging.

Important: The existing **log4j.properties** configuration is supported only for backward compatibility. After installing ECE 11.3 Patch Set 4, reconfigure all the default settings in the `ECE_home/occeserver/config/log4j2.xml` file to match your customized settings in the existing **Log4j.properties** file.

Enhanced Extension Points to Enrich and Filter All External Notifications

In previous releases, you could use the post-charging and post-update extension points to enrich and filter only few external notifications.

With this enhancement, you can use the post-charging and post-update extension points to enrich and filter all external notifications, excluding Advice of Charge (AOC) notifications.

You can add custom data to any external notification that is generated to provide additional data by using the post-update extension. For example, you can use the post-update extension to enrich external notifications generated for any updates post billing in BRM or for balance adjustments in BRM, such as spending limit notifications.

You can add custom data to any external notification that is generated post usage-charging in the charging flow by using the post-charging extension. For example, you can use the post-charging extension to enrich external notifications, such as diameter notifications, credit threshold notifications, or credit limit notifications. You can also filter out such external notifications that you do not want to be published to external systems.

For enriching or filtering external notifications by using the post-update extension, see "[ECE Extension Point for Post-Update to Enrich and Filter Out External Notifications](#)".

For enriching or filtering external notifications by using the post-charging extension, see the discussion about post-charging extension in *BRM Elastic Charging Engine Extensions*.

Sessions Can Now Be Charged Based on the Session Connect Time

By default, the session attempt time, which is the time the session is initiated, is used as the session start time for calculating charges for usage sessions. For example, when a customer initiates a call at 10:00:00AM and the call actually gets connected at 10:00:30 AM, 10:00:00 AM is considered as the call start time.

With this enhancement, you can configure ECE to use the session connect time, which is the time the session actually begins, as the session start time for calculating charges for usage sessions. You can do this by setting the **connectionTimeEnabled** entry in the `ECE_home/occeserver/config/management/charging-settings.xml` file to **true**.

To use the session connect time for calculating charges:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand **charging.reservationConfig**.
3. Expand **Operations**.
4. For each product that you offer, do the following:
 - a. Select **enabledOrDisableconnectionTime**.
 - b. Specify values for the following parameters:
 - productType**. Enter the name of the product defined in the ECE request specification data (for example, VOICE or SMS).
 - enableOrDisable**. Enter **true** to use the session connect time.
 - c. Click the **enabledOrDisableconnectionTime** button.

Summary of Fixes in ECE 11.3 Patch Set 4

Table 1–10 lists the bugs that were fixed in ECE 11.3 Patch Set 4 and provides a brief description of the resolution.

Table 1–10 Bug Fixes in ECE 11.3 Patch Set 4

SR Number	Bug Number	Description
3-12592163901	23220343	When a new product was purchased and if a high priority policy counter is added for the purchase, ECE was not sending a subscribe-notification-request (SNR). This has been fixed.
3-14579029591	25839811	In a multischema environment, when one instance of Customer Updater was started to load customer data, it was going into the usage processing state after loading the data. This was not allowing other instances of Customer Updater to load the customer data. This has been fixed.
3-14678315081	25887840	When a Sy session started by a Policy and Charging Rules Function (PCRF) node ended for some reason and failed over to an alternate peer PCRF node, the PCRF node sent a Diameter Peer Request (DPR) to ECE and ECE sent a Diameter Peer Answer (DPA). However, ECE did not send a SNR after sending the DPA to the PCRF node. This has been fixed.
3-14757682441	25956509	In a high-availability system, even when one instance of ECE failed, Rated Event Publisher could not publish the rated event to the Oracle NoSQL database data store if the grid rebalancing was in progress. This has been fixed.
3-14767047721	25985503	When the PCM_OP_CUST_MODIFY_CUSTOMER opcode was used, the credit profile information was not getting updated in ECE. This has been fixed.

Table 1–10 (Cont.) Bug Fixes in ECE 11.3 Patch Set 4

SR Number	Bug Number	Description
3-14796385281	25991876	When the ECE sync command was run to synchronize ECE configuration files, an error was displayed if the banner in the remote Secure Shell (SSH) server was enabled. This has been fixed.
3-14768147041	26000606	After billing, the customer details with discount sharing groups were removed from the cache. This has been fixed.
3-14863612201	26033102	When rating a prerated usage using a price override, the tax code was incorrectly populated as null. This has been fixed.
3-14788267451	26035463	When an account with cancelled products or discounts was cancelled, an error was displayed. This has been fixed.
3-14902282321	26076531	The rolling upgrade was failing for some specific scenarios. This has been fixed.
3-14914741101	26088220	Subscriber IDs for some of the subscribers participating in a closed user group were missing from the ECE cache, due to which the closed-user-group rule configured in ECE was not working for those subscribers. This has been fixed.
NA	26109656	Some BRM event fields with system generated values could not be customized. This has been fixed.
NA	26176858	The Rated Event Formatter custom plug-in could not access the application configuration data. This was impacting ECE implementation. This has been fixed.
NA	26186890	If the tax code and tax time values were not provided in a charge, the tax code and tax time were set as null in ECE. This has been fixed.
3-15003770091	26190324	Customer Updater was not starting after republishing the application configuration and updating ECE MBean attributes. This has been fixed.
3-15077149351	26263512	It was observed that the credit profile configuration with fixed thresholds as 0.0 was failing in ECE. This has been fixed.
3-15119349051	26271051	The tax time was incorrectly set as event time instead of using the taxation time configured in the charge offer and thus resulted in incorrect tax calculations. This has been fixed.

Table 1–10 (Cont.) Bug Fixes in ECE 11.3 Patch Set 4

SR Number	Bug Number	Description
3-15119738151	26271480	For some rating scenarios, there were differences in the rating between ECE and Pipeline Manager due to inherent differences in multi-rateable usage metric (RUM) rating evaluation. This has been fixed.
3-15119573491	26277726	The rating extensions in ECE did not have a similar functionality, such as the EVAL function, that is available in the pipeline discounting. This has been fixed.
3-15134780901	26283739	When loading customers from the BRM database, Customer Updater was failing with errors. This has been fixed.
3-15085623241	26331634	The discount impact category configured as pricing data was not populated in the rating period data. This has been fixed.
3-15181741181	26333062	It was not possible to the trace alteration configuration used for evaluating a specific rating period. This has been fixed.
NA	26370812	For currency credits, ECE was not considering the tax code in the charge offer if the tax code was not specified in the balance impact. This has been fixed.
3-15223735011	26375888	Cascading discounts in ECE were not evaluating the expressions as it was done in pipeline manager based solution. This has been fixed.
3-15283409493	26412528	When an alteration in ECE included both grant and consumption, the balance in BRM was not updated properly. This resulted in balance mismatch. This has been fixed.
3-15289774011	26448290	Oracle Communications Offline Mediation Controller ECE Distribution Cartridge was not receiving any response from ECE and thus could not process requests. This has been fixed.
3-15182173641	26481829	After a service update and rate plan change, the rating was failing for delayed usages. This has been fixed.

Documentation Updates in ECE 11.3 Patch Set 4

This section provides an overview of the documentation updates introduced in ECE 11.3 Patch Set 4.

The following changes have been made to the documentation:

- A new chapter with information about backing up and restoring ECE has been added in *BRM Elastic Charging Engine System Administrator's Guide*.

Known Problems in ECE 11.3 Patch Set 4

This section provides an overview of the known problems in ECE 11.3 Patch Set 4.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)

ECE 11.3 Patch Set 3 Release Notes

This section provides information about ECE 11.3 Patch Set 3.

New Features in ECE 11.3 Patch Set 3

This section provides an overview of the features introduced in ECE 11.3 Patch Set 3.

Additional Charging Result Attributes Can Be Used for Item Assignment

You can now use the following additional charging result (**CHARGING_RESULT_SPEC**) attributes for item assignment:

- **ORIG_ZONE_RESULT**
- **PRICING_NAME**
- **TIME_MODEL_NAME**
- **ZONE_MODEL_NAME**
- **EVALUATED_ZONE_MODEL_NAME**
- **EVALUATED_ZONE**
- **CHARGE_RATE_PLAN_NAME**

ECE uses these attributes to derive bill items for assigning balance impacts.

You can configure these fields in item type selectors in Pricing Design Center (PDC). For more information, see the discussion about using additional charging result attributes in *PDC Release Notes*.

Advice of Charge Notifications Now Include Quantity Details

ECE sends Advice of Charge (AoC) notifications to customers with the information about the charge for a service at the beginning of a session, during a session or at the end of a session. The AoC notifications now include information about the quantity used for charging, such as volume, usage duration, specific units, and so on.

The following is an example of the XML structure of the payload published for an AoC notification with usage duration:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Notification>
  <AdviceOfChargeNotification>
    <NotificationType>ADVICE_OF_CHARGE_NOTIFICATION_EVENT</NotificationType>
    <ChargeInfo>
      <PublicUserIdentity>2348044666646:TelcoLteVoice</PublicUserIdentity>
      <Consumptions>
        <BalanceElementId>566</BalanceElementId>
        <ConsumptionQuantity>529.20</ConsumptionQuantity>
        <ConsumptionUnit>Money {cur=NGN}</ConsumptionUnit>
      </Consumptions>
      <CurrentBalance>
        <BalanceElementId>566</BalanceElementId>
```

```

        <CurrentBalanceQuantity>-470.800</CurrentBalanceQuantity>
        <CurrentBalanceUnit>Money{cur=NGN} </CurrentBalanceUnit>
    </CurrentBalance>
</ChargeInfo>
<QuantityInformation>
    <QuantityInfo>
        <Name>DURATION</Name>
        <Quantity>100</Quantity>
        <Unit>Seconds</Unit>
    </QuantityInfo>
</QuantityInformation>
</AdviceOfChargeNotification>
</Notification>

```

For the XSD schema of the payload for the AoC notification, see the *ECE_home/occeserver/config/Notification.xsd* file.

Additional Data Storage Node Connections Can Be Configured for High Availability of Oracle NoSQL Database

In previous releases, you could configure only one data storage node connection for connecting to the Oracle NoSQL database. Therefore, when ECE processes were restarted, if the configured data storage node was down, ECE processes could not connect to any running data storage node.

With this enhancement, you can configure additional Oracle NoSQL database data storage node connections for use by ECE processes. This allows for failover and ensures high availability of the Oracle NoSQL database system. When ECE processes are restarted, if one of the configured data storage nodes goes down, ECE processes can connect to any other configured data storage node that is currently running.

To configure additional Oracle NoSQL database data storage node connections:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand **charging.connectionConfigurations.noSQLConnectioninstance**, where *noSQLConnectioninstance* is the name of the Oracle NoSQL database instance; for example, **noSQLConnection1**.
3. Expand **Attributes**.
4. Specify the value for the following attribute:
 - **dataStoreConnection**. Enter the *hostname:port* for connecting to the Oracle NoSQL database instance. The default is "localhost:5000" for connecting to a standalone Oracle NoSQL database instance. You can add additional data store connections separated by a comma. For example:

```
server1:5000,server2:6000,server3:7000
```

CER Message Now Contains a List of Configured Vendor IDs

Diameter peers exchange their identities and capabilities (such as product name, security mechanism, and so on) by using Capability-Exchange-Request (CER)/Capability-Exchange-Answer (CEA) Messages.

Diameter Gateway now reads all the attribute-value pairs (AVPs) configured (including custom AVPs) and populates all the Vendor IDs in the CER/CEA messages.

For information about the Diameter standard AVPs and Oracle custom AVPs used by Diameter Gateway for processing Gy, Sy, and Sh interface request types, see *BRM Elastic Charging Engine Diameter Gateway Protocol Implementation Conformance Statement*.

Configurable Log Location for Storing ECE Log Files

In previous releases, ECE was storing all the log files only in the default location, *ECE_home/occeserver/logs*.

With this enhancement, you can configure a new log location for ECE to store all the log files. If a location is not configured, ECE stores the log files by default in the *ECE_home/occeserver/logs* directory.

To configure a log location:

1. On the driver machine, open the *ECE_home/occeserver/config/ece.properties* file.
2. Search and uncomment the following system property:

```
#logDir =
```

3. Specify the path to the new log location:

```
logDir = ECE_log_path
```

where *ECE_log_path* is the absolute path to the directory that you want to use as your log location.

For example:

```
logDir = ECE_home/ece/logs
```

4. Save and close the file.

ECE Extension Point for Post-Update to Enrich and Filter Out External Notifications

ECE provides an extension point for post-update extensions in the updates processing flow (after receiving update requests and before publishing external notifications). You can use the post-update extension point to implement a custom logic to:

- Enrich external notifications. For example, you can add custom data to external notifications, such as spending limit notifications.
- Filter out external notifications that you do not want to be published to external systems. For example, when the billing is run, ECE generates subscribe-notification-request (SNR) notifications for all impacted resources. You can filter out unneeded SNR notifications and publish only required notifications to external systems.

To use the post-update extension, define the post-update extension's fully qualified class name in the *ECE_home/config/management/charging-settings.xml* file. You can define the class name by editing Extensions MBeans.

To define the class name for the post-update extension:

1. Access the ECE MBeans:

- a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand **charging.extensions**.
 3. Expand **Attributes**.
 4. Specify the value for the following attribute:
 - **postUpdateExtension**. Enter the fully qualified class name for the post-update extension. For example:
`oracle.communication.brm.charging.sdk.extensions.SamplePostUpdateExtension`

See the **SampleUpdateNotificationExtension** sample program in the ECE SDK (`ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions` directory) for an example of how to implement a custom logic by using the post-update extension.

See the discussion about sample extensions in *BRM Elastic Charging Engine Extensions* for information on how to use **SampleUpdateNotificationExtension**.

ECE Now Supports Selective Loading of Data from BRM into ECE

You can now load customer data and product cross-reference data selectively from BRM into ECE. With this enhancement:

- Running **Customer Updater** in the **selectiveMigrationMode** mode performs the initial loading of customer data and product cross-reference data only for the product offerings (services and the corresponding pricing data) that are currently stored in the ECE cache, which are loaded from PDC into ECE.

If you are using ECE for usage rating, when you migrate data selectively from BRM into PDC, do the following in ECE:

1. Configure **Customer Updater** to run in the **selectiveMigrationMode** mode by setting the **selectiveMigrationMode** attribute in the `ECE_home/config/management/migration-configuration.xml` file to **true**.

See "[Configuring Customer Updater to Load Data Selectively](#)".

2. Load the selectively migrated pricing data (including services) from PDC into ECE by running Pricing Updater.

See the discussion about loading pricing data into ECE in *BRM Elastic Charging Engine Implementation Guide*.

3. Load the corresponding customer data and product cross-reference data from BRM into ECE by running Customer Updater.

See the discussion about loading data from BRM into ECE in *BRM Elastic Charging Engine Implementation Guide*.

- Running the **Customer Loader** utility with the **-loadCrossRefData** parameter loads the new or modified set of product cross-reference data from BRM into ECE for the schema that you specify. The **Customer Loader** utility loads the product cross-reference data only for the product offerings that are stored in the ECE cache when the utility is run with the **-loadCrossRefData** parameter. For more information, see "[Loading Product Cross-Reference Data with the customerLoader Utility](#)".

Important: In production environment, run the **Customer Loader** utility only with the **-loadCrossRefData** or **-incremental** parameter.

In testing or non-production environment, run the **Customer Loader** utility without any parameters.

Configuring Customer Updater to Load Data Selectively

To configure Customer Updater to load data selectively:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand **migration.loader**.
3. Expand **Attributes**.
4. Set the **selectiveMigrationMode** attribute to **true**.

Loading Product Cross-Reference Data with the customerLoader Utility

To load product cross-reference data with the **customerLoader** utility:

1. Start ECC:

```
./ecc
```
2. If your charging server nodes are not running, start them and load configuration data and pricing data:

```
start server
start configLoader
start pricingLoader
start CustomerUpdater
```
3. Run the following command, which loads the cross-reference data from the specified customer updater schema:

```
start customerLoader -loadCrossRefData customer_updater_schema_name
```

where *customer_updater_schema_name* is the schema name specified for Customer Updater in the **connectionconfiguration** section of the *ECE_home/config/management/charging-settings.xml* file.

For example:

```
start customerLoader -loadCrossRefData customerUpdater1
```

For loading cross-reference data from multiple schemas, run the **customerLoader** utility for each schema; for example, *customerUpdater1*, *customerUpdater2*, and so on.

KeepAlive Enabled for Connections Between BRM Connection Manager and EM Gateway

BRM Connection Manager (CM) and External Manager (EM Gateway) use a pool of connections to send/receive requests. These connections are activated during BRM Connection Manager startup.

In previous releases, when there were no requests exchanged between CM and EM Gateway for a defined period of time, the idle connections were closed by the firewall.

To prevent this, the KeepAlive option is now enabled on the listening sockets. This allows EM Gateway to use the operating system's KeepAlive settings.

Important: Before starting EM Gateway, ensure that the KeepAlive interval (**tcp_keepalive_interval**) configured for the operating system does not exceed the idle connection timeout configured in the firewall.

By default, the KeepAlive interval for the operating system is set to 7200 seconds. You must reduce this interval so that it does not exceed the idle connection timeout. See your operating system documentation for information on reducing the KeepAlive interval.

By default, EM Gateway is enabled to use the operating system's KeepAlive settings. You can prevent EM Gateway from using the operating system's KeepAlive settings by setting the **socketKeepAlive** entry in the **emGatewayConfigurations.Instance_Name** (where *Instance_Name* is the name of the EM Gateway instance; for example, *emGateway1*) section of the *ECE_home/config/management/charging-settings.xml* file to 0.

New Reason Code Introduced to Indicate Rating Failure

In previous releases, ECE returned a failure message with the NO_RATED_QUANTITY reason code in the following cases:

- If the graph for the rateable usage metric (RUM) configured for rating could not be found in the path specified
- If the rating failed due to insufficient balance after reverse rating. See the discussion about reverse rating in *BRM Elastic Charging Engine Concepts*.

With this enhancement, ECE returns a failure message with a new reason code, NO_RATING_GRAPH_CONFIGURED, if the graph for the RUM configured for rating could not be found in the path specified. However, in multiple RUMs scenario, if the graph could not be found for any of the RUMs configured for rating, ECE uses the NO_RATED_QUANTITY reason code to indicate the rating failure.

By default, if the graph is missing, ECE considers it as an error and returns a failure message with the `NO_RATING_GRAPH_CONFIGURED` reason code. However, you can configure ECE to not consider this as an error by setting the `treatNoRatingGraphAsError` attribute in the `ECE_home/config/management/charging-settings.xml` file to `false`. When set to `false`, ECE reports the `NO_RATING_GRAPH_CONFIGURED` reason code in its response and continues rating.

To skip missing graph and continue rating:

1. Access the ECE MBeans:
 - a. Log on to the driver machine, which is the machine on which you installed ECE.
 - 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.
2. Expand `charging.server`.
3. Expand **Attributes**.
4. Set the `treatNoRatingGraphAsError` attribute to `false`.

Summary of Fixes in ECE 11.3 Patch Set 3

Table 1–12 lists the bugs that were fixed in ECE 11.3 Patch Set 3 and provides a brief description of the resolution.

Table 1–11 Bug Fixes in ECE 11.3 Patch Set 3

SR Number	Bug Number	Description
3-13366583201	24708085	After failover, the running instance of Diameter Gateway could not handle SNR notifications sent by the Diameter Gateway instance that went down. This has been fixed.
3-13436631451	24808918	Customer Updater failed to load the data of all customers. This has been fixed.
3-13456098661	24829099	The primary and secondary currency information was not synchronized correctly with ECE. This has been fixed.
3-13457662791	24829344	Purchasing an add-on item type bundle was failing. This has been fixed.
NA	24960515	Custom plug-ins were not able to access rating period and correlation rating period identifier information. This has been fixed.

Table 1–11 (Cont.) Bug Fixes in ECE 11.3 Patch Set 3

SR Number	Bug Number	Description
3-13648314078	25102947	ECE was not supporting tax jurisdictions. Tax jurisdiction details were not added to the tax rating period. This has been fixed.
3-13651362851	25132832	During the billing process, an SNR notification was getting generated erroneously while renewing the monthly bundle. This has been fixed.
3-13699502809	25152139	EM Gateway was generating an error when a service was created without any rating products. This has been fixed.
3-13701360011	25158929	JMS event creations were returning errors. This has been fixed.
3-13715937671	25160960	Installation of ECE 11.3.0.1.1 patch set had some issues on the Solaris platform. This has been fixed.
3-13692386931	25169963	JMS Connection issues were logged incorrectly with log severity as "DEBUG" instead of "WARNING." This has been fixed.
3-13766115221	25223609	After a restart of Diameter Gateway, the log level could neither be increased nor decreased through JMX editor. This has been fixed.
3-13894996311	25308131	During rating, the NOT (!) operator was not evaluated, due to which balance impacts were incorrect. This has been fixed.
3-13883180051	25338661	Discounts that were configured for the parent service were not applied during charging. This has been fixed.
3-13699502809	25339876	EM Gateway generated an error when a service was created without any rating products. This has been fixed.
3-13810434131	25365361	In ECE, only event time zone was used to do rating. There was no provision to override it with the account time zone. This has been fixed.
3-13087195451	25411305	The rounded beat quantity was not considered for rating. This has been fixed.
3-14050638941	25418707	After restart, Diameter Gateway was not processing SNR notifications for the existing session, even though the notifications were present in the JMS queue. This has been fixed.

Table 1–11 (Cont.) Bug Fixes in ECE 11.3 Patch Set 3

SR Number	Bug Number	Description
3-14036123431	25472353	In an ECE deployment, when there was no data exchanged between BRM and EM Gateway for a configured period of time, the connection was closed by the firewall. This has been fixed. See " KeepAlive Enabled for Connections Between BRM Connection Manager and EM Gateway ".
3-14251788740	25523084	For derived events, ECE did not evaluate discounts defined at the base event class level. For example, a discount configured at the level of EventDelayedSession was not applied for usage corresponding to EventDelayedSessionGprs although EventDelayedSession was the parent class of EventDelayedSessionGprs. This has been fixed.
3-13467263471	25651613 25651614	The AoC notification payload did not contain usage duration. This has been fixed. See " Advice of Charge Notifications Now Include Quantity Details ".
3-13607263341	25658871	EM Gateway returned an error when the value of the account-level Extended Rating Attribute (ERA) was null. This has been fixed.
3-14374231121	25664382	The Rated Event Formatter (REF) abruptly stopped when processing event data and required a restart to continue processing. This has been fixed.
3-13203395071	25700191	There was an issue in splitting a charge based on the time model. This has been fixed.
3-14579159551	25810259	In a multischema environment, when BRM Connection Manager, which was connected to the secondary schema, was used to manage subscriptions for an account residing in the primary schema, EM Gateway returned an error. This has been fixed.
3-14620013961	25825341	When using the post-rating extension, mutation on the block-level fields was failing. This has been fixed.
3-14614317881	25832260	Creation of account-level sharing groups was failing, while other types of sharing groups such as subscription sharing groups were created successfully. This has been fixed.

Table 1–11 (Cont.) Bug Fixes in ECE 11.3 Patch Set 3

SR Number	Bug Number	Description
3-14628754901	25839909	Certain discount expressions for charging were not supported in ECE. This has been fixed.
3-14315807911	25877762	Rating failed with error code NO_QUALIFIED_CHARGE_OFFERS for a charge offer having a Value Map configuration. This has been fixed.
3-14576303541	25911704	For events rated in ECE, not all the corresponding balance impact information (for example, RATE_TAG) was persisted in the EVENT_BAL_IMPACTS_T table in the BRM database. This has been fixed.

Known Problems in ECE 11.3 Patch Set 3

This section provides an overview of the known problems in ECE 11.3 Patch Set 3.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)

ECE 11.3 Patch Set 2 Release Notes

This section provides information about ECE 11.3 Patch Set 2.

New Features in ECE 11.3 Patch Set 2

This section provides an overview of the features introduced in ECE 11.3 Patch Set 2.

ECE Can Now Process Delayed Usage Requests for the Current Accounting Cycle

In the previous releases, to calculate usage charges and assign bill items, ECE processed only usage requests received during the same accounting cycle in which the usage occurred. ECE did not process the usage requests received after the end of the accounting cycle in which the usage occurred (called delayed usage requests).

With this enhancement, if delayed billing is configured in Oracle Communications Billing and Revenue Management (BRM), you can configure ECE to process delayed usage requests for the accounting cycle in which the usage occurred if the usage requests are received within the delay tolerance interval that you specify. This extends the item assignment for that accounting cycle by the specified interval so that associated rated events are assigned to the current open bill item. Delayed usage requests received after the specified interval are processed for the next accounting cycle, and the associated rated events are assigned to the next open bill item.

For more information, see the discussion about processing delayed usage requests in *BRM Elastic Charging Engine Concepts*.

ECE Can Now Determine Rates Based on Both the Calling and the Called Customers' Data

The pre-rating extension now provides access to the profile data of the called customer (irrespective of the profile selected) and to the purchased charge offerings data and purchased alteration offerings data of both the calling customer and the called

customer. You can use this data to determine rates for sessions by implementing a custom logic using the pre-rating extension. For example, you can implement a custom logic to modify the usage request to apply a special rate or discount (such as a birthday discount) for a call session based on the extended rating attributes of both the calling customer and the called customer.

See the **SamplePreRatingExtension** sample program in the ECE SDK (*ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions* directory) for an example of how to implement a custom logic by using the pre-rating extension.

For more information on the pre-rating extension, see *BRM Elastic Charging Engine Extensions*.

ECE Can Now Override Tax Rates or Assign Bill Items Based on the Extensions Data

ECE extensions now provide access to tax codes and custom item types. You can modify tax codes and custom item types by implementing custom logic using the ECE extensions. This allows you to override tax rates or assign bill items to event balance impacts for the rating period based on the data accessible through the ECE extensions (for example, called ID, time of call, or time zone).

To implement a custom logic to modify tax codes, use the post-rating extension. To implement a custom logic to modify custom item types, use the new Rating extension. See the **SamplePostRatingExtension** and **SampleRatingExtension** sample programs in the ECE SDK (*ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions* directory) for an example of how to implement a custom logic by using the post-rating extension.

For more information on the ECE extensions and the accessible data, see *BRM Elastic Charging Engine Extensions*.

For information on the extension APIs that are used by the post-rating extension and the rating extension for implementing the custom logic, see the documentation for **oracle.communication.brm.charging.extensions.client.PostRatingExtensionContext** and **oracle.communication.brm.charging.extensions.client.RatingExtensionContext** in *BRM Elastic Charging Engine Java API Reference*.

ECE Can Now Use Shared Profiles to Determine a Price

ECE can now apply pricing based on a customer's participation in a profile sharing group, which is a group that enables an account's service to share a profile with other accounts or services. A profile stores extended rating attributes (ERAs) or other types of information about an account.

You create the profile sharing group in BRM. Customer Updater loads the profile sharing group data from BRM into the ECE database.

For more information, see the discussion about working with profile sharing groups in *BRM Managing Customers*.

Enhanced ECE Charging Functionality

ECE charging functionality has been enhanced to override the price specified in the product offerings at run time. To override the price, you create a pricing XML file with dynamic tags and import the file into the PDC database by using the **ImportExportPricing** utility. Dynamic tags are the XML elements that are used for overriding the value of the pricing attributes, such as price, incrementStep, lowerBound, and UpperBound. PDC provides a sample XML file in the *PDC_home/apps/Samples/Examples* directory.

Note: Ensure that you create a unique dynamic tag across multiple charge offers since the tags are not scoped to charge offers at run time.

You can use the dynamic tags to override the default value of the pricing attributes by implementing custom logic using the pre-rating extension. When you implement the custom logic, the overridden values are populated in the event profile map in the payload of the request specification file. ECE uses these values to determine the price when processing a usage request.

For more information on the pre-rating extension, see *BRM Elastic Charging Engine Extensions*.

For information on the extension APIs that are used by the pre-rating extension for implementing the custom logic, see the documentation for **oracle.communication.brm.charging.extensions.client.PreRatingExtensionContext** in *BRM Elastic Charging Engine Java API Reference*.

Support for Using Rounding Rules Configured in PDC to Round the Charging Results

In the previous releases, ECE used only the rounding rules configured in ECE to round the charging results for each processing stage (such as charging, alteration, and taxation). The same rounding rules were used for each processing stage.

With this release, ECE uses the rounding rules configured in PDC to round the charging results for each processing stage. The rounding rules can be different for each processing stage. If a rounding rule is not configured in PDC, ECE uses the rounding rules configured in ECE to round the charging result.

For more information, see the discussions about rounding the charge results in *BRM Elastic Charging Engine Implementation Guide*.

Support for Using Default System Currency for Charging

With this release, you can configure ECE to use a default system currency for charging subscribers. During rating, ECE uses the subscriber's primary currency or the secondary currency. If the currency used in the rate plans does not match the subscriber's primary or secondary currency, ECE uses the default system currency, US dollars.

For more information, see the section about configuring default system currency in *BRM Elastic Charging Engine System Administrator's Guide*.

Summary of Fixes in ECE 11.3 Patch Set 2

[Table 1–12](#) lists the bugs that were fixed in ECE 11.3 Patch Set 2 and provides a brief description of the resolution.

Table 1–12 Bug Fixes in ECE 11.3 Patch Set 2

SR Number	Bug Number	Description
3-12745189011	23507496	Customer Updater was getting into an infinite loop and was consuming a lot of memory. This has been fixed.
3-13136715551	24554814	Customer Updater was getting into the idle state while loading data. This has been fixed.
3-13247973031	24558941	Setting event time stamps in the Sy messages during testing was not possible. This has been fixed.
3-13206140491	24561497	When a product was canceled in BRM, the validity updates to the balances were not synchronized with ECE. This has been fixed.
3-13203395071	24705052	The Spending Limit Request (SLR) was returning all the available counters rather than the applicable counters. This has been fixed.
3-13259991233	24706341	When the customer data in the ECE cache was overridden, some data was lost. This has been fixed.
3-13366583201	24708085	When configured in the high availability mode, notifications were not processed by Diameter Gateway after a failover. This has been fixed.
3-13436631451	24808918	Customer Updater was failing with a null pointer exception while loading data. This has been fixed.
3-13456098661	24829099	The primary and secondary currency data from BRM was not getting synchronized with ECE. This has been fixed.
3-13457662791	24829344	Customer Updater failed to update the data in ECE when a bundle with an item type offer was purchased. This has been fixed.
3-13468287671	24931236	ECE did not process old usage requests for the period in which the associated telephone numbers were active. This has been fixed.

Table 1–12 (Cont.) Bug Fixes in ECE 11.3 Patch Set 2

SR Number	Bug Number	Description
3-13607263341	25101265	When values for account-level extended rating attributes were missing or null, EM Gateway was throwing an error. This has been fixed.
3-13619630111	25107601	The original amount and resource ID fields in an event's bal_impacts array were not getting updated properly. As a result, the original amount and resource fields could not be altered for that event by using a post-rating or post-charging extension. This has been fixed.
3-13701360011	25175598	Errors were encountered when messages were published to the JMS queue. This has been fixed.

Known Problems in ECE 11.3 Patch Set 2

This section provides an overview of the known problems in ECE 11.3 Patch Set 2.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)

ECE 11.3 Patch Set 1 Release Notes

This section provides information about ECE 11.3 Patch Set 1.

New Features in ECE 11.3 Patch Set 1

This section provides an overview of the features introduced in ECE 11.3 Patch Set 1.

ECE Credit Threshold and Credit Limit Breach Notifications Now Include Additional Information

When a customer's balance breaches a credit threshold value or a credit limit, ECE sends a notification with information about the breach. ECE sends the following types of credit threshold or credit limit breach notifications:

- Threshold breach notifications
- Aggregated threshold breach notifications
- Credit ceiling breach notifications
- Credit floor breach notifications

The above notifications now include additional information about the breach: the alert type, the reason for the breach, and the type of operation. For Usage operations, the notifications include the subtype. For sample notifications, see the discussion about configuring notifications for charging in *BRM Elastic Charging Engine Implementation Guide*.

Optionally, you can add information such as calling number, called number, event type, and balance group to the above notifications by using the post-charging extension. For more information, see *BRM Elastic Charging Engine Extensions*.

You can also add the additional information to the header of the above notifications by configuring the notification header attributes using a JMX editor, such as JConsole. For more information, see the discussion about configuring notifications for charging in *BRM Elastic Charging Engine Implementation Guide*.

ECE Now Supports Sending Rated Events to a Non-BRM System

In previous releases, rated events could only be processed in the CDR format and sent to BRM.

With this release, you can process rated events in different formats by using a custom plug-in. This allows you to send rated events in a format required by any external system. For example, you can send rated events in the JSON format to a data warehouse for report and data analysis purposes. To implement a custom plug-in, use the **SampleRatedEventFormatterCustomPlugin** sample program in the ECE SDK (*ECE_home/occesdk/source/oracle/communication/brm/charging/sdk/extensions* directory).

See the following for information about implementing and using the custom plug-in:

- The discussion about the sample programs in *BRM Elastic Charging Engine Implementation Guide* for information on how to compile and run the **SampleRatedEventFormatterCustomPlugin** sample program
- The discussion about the custom plug-in API, installing and configuring Rated Event Formatter, and configuring the *BrmCdrPluginDirect* plug-in in *BRM Elastic Charging Engine Implementation Guide*

ECE Now Supports the BRM Gateway Suspense Queue Configuration

ECE now supports a suspense queue to collect any failed updates through BRM Gateway.

Sometimes, update requests do not successfully move from ECE to BRM. To prevent data loss, you can now collect the failed update requests sent through the BRM Gateway by configuring a JMS suspense queue. You can use this queue to store the failed updates until they can be reprocessed.

See the following for information about collecting and moving the failed events:

- The discussion about creating a suspense queue for the BRM Gateway in *BRM Elastic Charging Engine Installation Guide*
- The discussion about configuring the BRM Gateway and configuring a suspense queue for BRM Gateway in *BRM Elastic Charging Engine Implementation Guide*
- The discussion about troubleshooting the failed update requests from ECE in *BRM Elastic Charging Engine System Administrator's Guide*

ECE Now Supports Processing Offline Charging Requests for Accounts for Which Billing Has Been Delayed

When ECE receives an offline charging request for an account for which billing has been delayed, it does the following:

- Rejects the offline charging request with the BILLING_DUE reason code
- Sends a billing notification to BRM to trigger partial billing

ECE now includes the following enhancements to process the offline charging request for the account:

- A new attribute, **requestMode**, has been introduced in the usage request to determine that a request is an offline charging request.

If you use Oracle Communications Offline Mediation Controller as your network mediation software for offline charging, see the discussion on creating and configuring the Elastic Charging Engine distribution cartridge node in *Oracle Communications Offline Mediation Controller Elastic Charging Engine Cartridge Pack User Guide* for information on configuring the request mode for ECE.

- Customer Updater and External Manager Gateway have been enhanced to load the billing state information from the BRM database into ECE. This allows ECE to determine whether the billing for an account has been delayed or completed.
- A new reason code, **BILLING_DUE**, has been included in the response message for these offline charging requests.

ECE does not directly handle suspense management for offline charging requests. If you are using Offline Mediation Controller as your network mediation software for offline charging, configure Offline Mediation Controller for handling suspense management for these requests. For more information, see *Oracle Communications Offline Mediation Controller Suspending and Recycling Call Detail Records User's Guide*.

See the discussion about billing notifications in *BRM Elastic Charging Engine Concepts* for more information.

Enhanced ECE External Notifications

The following ECE external notifications now include operation type and its corresponding subtypes to indicate when the notification is triggered:

- Advice of Charge (AoC) notification
- Re-authorization Request (RAR) notification
- Spending Limit notification
- Subscriber Preferences notification
- Top-Up notification

For sample notifications, see the discussion about configuring notifications for charging in *BRM Elastic Charging Engine Implementation Guide*.

Support for Changing ECE's Current Time and Date to Test ECE

You can now change ECE's current time and date to test time-sensitive functions associated with Rated Event Formatter and Diameter Gateway in ECE. You can change the time and date by using a JMX editor.

For more information on changing ECE's current time and date to test ECE, see the discussion of ECE testing tools in *BRM Elastic Charging Engine Implementation Guide*.

Summary of Fixes in ECE 11.3 Patch Set 1

[Table 1–13](#) lists the bugs that were fixed in ECE 11.3 Patch Set 1 and provides a brief description of the resolution.

Table 1–13 Bug Fixes in ECE 11.3 Patch Set 1

SR Number	Bug Number	Description
3-12189891421	22812011	Concurrent access to the accounts that share the same balance was failing with errors. This has been fixed.
3-12545233491	23201547	If you created a custom business event (for example, CustModify), which included a preconfigured business event (for example, ProductModifyEvent), triggering the custom business event was not considering the preconfigured business event. This has been fixed.
3-12557104791	23201553 23201556	The ProductPurchaseEvent event was not getting updated properly in ECE. This has been fixed.
3-12514667881	23208355 23208356	While updating the customer balances, Customer Updater failed with errors when a decimal value range was encountered. This has been fixed.
3-12660130741	23260275	Customer-level discount sharing was failing with errors. This has been fixed.
3-12554866671	23289852	When a billing-time discount was configured, the tax calculation for discounting was incorrect. This has been fixed.
3-12733546691	23333898	The threshold breach notification data had exponential values for the threshold amount and it was impossible to identify whether the breach was due to fixed or percentage threshold. This has been fixed.
3-12523309331	23480353 23480366	When ECE nodes were restarted, the pricing loader utility was taking a longer time to load data into the ECE cache. This has been fixed.
3-12726200401	23482770	ECE was throwing exceptions while rating the events for the accounts for which distribution offers (charge sharing offers) were configured. This has been fixed.
3-12702149501	23538076	ECE was considering the quantity (cascading) discount mode for applying discount. This has been fixed.
3-12758588241	23554720	Subscribe-Notifications-Requests (SNRs) were not getting generated for voucher top-up. This has been fixed.
3-12592944241	23577007 23577011	While processing the update requests, Customer Updater rejected events such as AccountStatusUpdate and UpdateServicesEvent with an error. This has been fixed.

Table 1–13 (Cont.) Bug Fixes in ECE 11.3 Patch Set 1

SR Number	Bug Number	Description
3-12593681301	23579599 23621595 23621596	When ECE nodes were restarted, some unnecessary metadata and pricing and configuration data were loaded into BRM. This was impacting the product catalog maintained in CRM. This has been fixed.
3-10690047251	23593123 23593126	After a telephone number of a subscriber was changed to a new number, if any usage request was received with the new number for a date earlier than the date the telephone number was changed, ECE processed the usage request instead of displaying an error. If the ECE server was restarted after changing the telephone number, ECE was not processing the usage request with the old telephone number even if the request was for a date earlier than the date the telephone number was changed. This has been fixed.
3-12647332091	23621579 23621581	When a product was canceled, an unexpected threshold breach notification was generated. This has been fixed.
3-12897304221	23748630	Generation of custom notifications by using the post-charging extension was failing with an error. This has been fixed.
3-13094525201	24384621	When partial billing was enabled for an account, EM Gateway was throwing an error at the time of billing. This has been fixed.
3-12824633391	24408141 24408148	EM Gateway was returning an invalid error code for some of the errors. This has been fixed.
3-13165129691	24445290	If an account was created and later the service and product were added to the account, Spending-Limit-Request (SLR) for that account was ignored with an error. This has been fixed.
3-12557552201	24477811	Active or consumed reservations were not removed even after the specified expiry time. This has been fixed.

Known Problems in ECE 11.3 Patch Set 1

This section provides an overview of the known problems in ECE 11.3 Patch Set 1.

See the following for more information:

- [SLM Feature Configuration Is Not Automatically Loaded into ECE](#)
- [Rolling Upgrade Does Not Work for Upgrading to ECE 11.3 Patch Set 1](#)

Rolling Upgrade Does Not Work for Upgrading to ECE 11.3 Patch Set 1

A *rolling upgrade* using the `rollingUpgrade` command does not work for upgrading to ECE 11.3 Patch Set 1.

There is no workaround for upgrading to ECE 11.3 Patch Set 1 while existing installation is running. You must stop all ECE nodes of your existing installation, restore your ECE system, and then start all ECE nodes of your existing installation. For more information, see the discussion about stopping and restoring your ECE system in *BRM Elastic Charging Engine Installation Guide*.

ECE 11.3 Release Notes

This section provides information about ECE 11.3.

New Features in ECE 11.3

This section provides an overview of the features introduced in ECE 11.3.

ECE Now Has RADIUS Gateway to Support RADIUS Protocols

ECE now supports RADIUS protocols. ECE implements RADIUS protocols by using RADIUS Gateway. You use RADIUS Gateway to process authentication and accounting requests when your customers use terminal servers or Network Access Server (NAS) to connect to ECE.

RADIUS Gateway is an online charging system (OCS) front-end server for the Oracle Communications Billing and Revenue Management (BRM) system. It is a component of ECE. It acts as a RADIUS server and presents ECE to the network as a RADIUS application. It translates RADIUS requests received from RADIUS clients (terminal servers or NAS) into ECE Java API requests. It translates the response from the Elastic Charging Server back into RADIUS responses and responds back to the requesting RADIUS client.

RADIUS Gateway is included in the ECE Server installation, and you can deploy RADIUS Gateway nodes into the ECE cluster the same way you deploy other ECE nodes. RADIUS Gateway has ready-to-use example configuration files to facilitate implementation.

Important: RADIUS Gateway is an optional component that requires a separate license.

For more information about RADIUS Gateway, see the discussion about the ECE system architecture in *BRM Elastic Charging Engine Concepts*.

For more information about deploying RADIUS Gateway nodes into the ECE cluster, see post-installation tasks in *BRM Elastic Charging Engine Installation Guide*.

For more information about configuring RADIUS Gateway to receive RADIUS requests, mapping RADIUS network attributes to event attributes, and customizing the RADIUS data dictionary in ECE, see the discussion about configuring and customizing RADIUS Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For more information about using RADIUS Gateway for processing authentication requests, see the discussion about authentication using RADIUS Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For more information about using RADIUS Gateway for processing accounting requests, see the discussion about accounting using RADIUS Gateway in *BRM Elastic Charging Engine Implementation Guide*.

For information about starting and stopping RADIUS Gateway and troubleshooting failed request processing in RADIUS Gateway, see *BRM Elastic Charging Engine System Administrator's Guide*.

For information about the extension points available for processing RADIUS requests, see the discussion about the extension points that process RADIUS requests in *BRM Elastic Charging Engine Extensions*.

For information about the messages and attribute-value pairs used by RADIUS Gateway for processing requests from RADIUS clients, see *BRM Elastic Charging Engine RADIUS Gateway Protocol Implementation Conformance Statement*.

ECE Now Supports Disaster Recovery

Disaster recovery helps resume processing of ECE requests when the infrastructure at the production site breaks down. ECE supports the following disaster recovery configurations:

- **Active-Cold Standby.** A disaster recovery system that consists of an active production site and an idle backup site at a remote location. When the production site goes down, you need to bring up the backup site to full operational capability.
- **Active-Hot Standby.** A disaster recovery system that consists of an active production site and an active backup site at a remote location. When the production site goes down, the ECE requests are diverted from the production site to the backup site.
- **Segmented Active-Active.** A disaster recovery system that consists of two or more active production sites at remote locations, which concurrently processes ECE requests for a specific set of customers. When one of the production site goes down, the requests from that site are diverted to the other sites.

For more information, see the discussion about disaster recovery in *BRM Elastic Charging Engine System Administrator's Guide*.

BRM-to-ECE Data Updates Are Now Synchronized in Real Time

When the BRM server performs customer management and billing transactions, it stores the results in the BRM database. To enable ECE to rate usage events properly, all customer data updates made in the BRM database must also be made in the ECE cache.

In the previous releases, such updates were published to ECE asynchronously. In that mode, the ECE cache was updated after the updates were saved to the BRM database and the transaction was closed. The process used the Account Synchronization Data Manager (DM), an Oracle Advanced Queuing (AQ) database queue, and Customer Updater. Because delays could occur when Customer Updater dequeued the updates, a lag could exist between the time the BRM database was updated and the time the ECE cache was updated. During the lag, the BRM and ECE data were unsynchronized. In addition, because the updates had already been saved to the BRM database, the BRM and ECE data would be unsynchronized if the ECE cache update failed. If ECE used the unsynchronized data to rate usage events, the events might be incorrectly rated.

In this release, such updates are published to ECE synchronously. In this mode, both the database and the cache updates occur within the original 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. Because both the database and the cache are updated within the same transaction, no lag time occurs, and the BRM and ECE data remain synchronized whether the cache update succeeds or fails. Maintaining the

synchronization of the BRM and ECE data preserves the integrity within the BRM system of calculations ECE makes based on that data. This process uses EM Gateway. See the discussion about synchronizing BRM and ECE customer data in *BRM Elastic Charging Engine Concepts*.

ECE Can Now Generate Midsession Rated Events

To enable BRM to bill for usage during online network sessions, ECE must generate rated events and send them to the BRM server.

In previous releases, ECE generated a rated event for a network session only when the session ended with a Diameter terminate operation. Some sessions, however, such as data streaming, last days, weeks, or months. Many network operators do not want to wait until the end of such lengthy sessions to bill for the part of the session that subscribers have already consumed.

Therefore, in this release, you can configure ECE to generate a rated event for any Diameter update operation that occurs during a network session. The session must meet or exceed criteria you specify, such as session duration, download quantity, or time of day. These midsession rated events enable BRM to bill incrementally for usage during long network sessions, preventing large amounts of unrated usage and unrecognized revenue from accumulating. Midsession rated events also enable operators to show subscribers their running balance throughout a session.

See the discussions about midsession rated events in *BRM Elastic Charging Engine Concepts* and *BRM Elastic Charging Engine Implementation Guide*.

ECE Supports Credit Control Requests Without Multiple-Service Credit Control Attribute-Value Pairs

In previous releases, ECE reported an error if the Diameter Credit Control request (CCR) did not contain any multiple-service credit control (MSCC) AVPs.

With this release, ECE reports an error only if the subscriber ID is not present in the CCR. This allows the users to send CCRs without MSCC AVPs for subscriber authentication.

For more information, see the discussion about network integration for online charging using Diameter Gateway in *BRM Elastic Charging Engine Implementation Guide*.

ECE Now Supports Customer-Level Charge Offers, Alteration Offers, and Distribution Offers

In previous releases, ECE supported only product-level charge offers, alteration offers, and distribution offers.

With this release, ECE supports customer-level charge offers, alteration offers, and distribution offers for charging, pricing, and rerating.

For more information, see the discussion about creating pricing data for the ECE runtime environment in *BRM Elastic Charging Engine Implementation Guide*.

ECE Now Supports Incremental Loading of Customer Data

You can now load customer data from BRM into ECE incrementally. With this enhancement:

- Running the **Customer Loader** utility with the **incremental** parameter extracts customer data, segregates it into different batches, and incrementally loads the customer data from BRM into ECE.

Note: In production environment, run the **Customer Loader** utility with this parameter.

- Running the **Customer Loader** utility without any parameters loads the sample customer data from XML files into ECE.

Note: In testing or non-production environment, run the **Customer Loader** utility without any parameters.

For more information, see the discussion about incremental loading of customer data in *BRM Elastic Charging Engine Implementation Guide*.

Documentation Additions

This section provides an overview of the documentation updates introduced in ECE 11.3.

The following changes have been made to the documentation:

- *BRM Elastic Charging Engine RADIUS Gateway Protocol Implementation Conformance Statement* is added. *RADIUS Gateway Protocol Implementation Conformance Statement* lists the RADIUS messages and AVPs used by RADIUS Gateway for processing RADIUS and RADIUS Accounting interface request types. Prerequisite reading for *RADIUS Gateway Protocol Implementation Conformance Statement* are *BRM Elastic Charging Engine Concepts* and *BRM Elastic Charging Engine Implementation Guide*.
- *BRM Elastic Charging Engine System Administrator's Guide* contains a utility page for the **gridSync** utility.

Known Problems in ECE 11.3

This section provides an overview of the known problems in ECE 11.3.

SLM Feature Configuration Is Not Automatically Loaded into ECE

SR Number: Not applicable

Bug Number: 23031547

When you configure the Service Lifecycle Management (SLM) feature on the BRM server, the SLM feature configuration is not automatically loaded into ECE. As a result, the subscriber lifecycle states are not enabled in ECE.

To work around this problem, do the following in ECE before you start the charging server nodes:

1. Verify that the charging server nodes are not running.
2. Open the *ECE_home/occeserver/config/management/charging-settings.xml* file.
3. Locate the **lifecycleStateMappingConfiguration** section.
4. Do one of the following:

- To enable the default subscriber lifecycle states, comment out the following lines by using the pound (#) symbol:

```
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="101" stateName="PREAMTIVE"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="102" stateName="ACTIVE"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="103" stateName="RECHARGE_
ONLY"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="104" stateName="CREDIT_
EXPIRED"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="105" stateName="DORMANT"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="106" stateName="FRAUD_
INVESTIGATED"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="107" stateName="SUSPENDEED"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.product state.LifecycleState"state="108" stateName="CLOSED"/>
```

- To enable the custom subscriber lifecycle states, comment out the following lines by using the pound (#) symbol:

```
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="10100" stateName="ACTIVE"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.productstate.LifecycleState"state="10102"
stateName="SUSPENDEED"/>
<lifecycleStateconfig-class="oracle.communication.brm.charging.appconfigura
tion.beans.product state.LifecycleState"state="10103" stateName="CLOSED"/>
```

5. Save and close the file.
6. On the machine on which you have ECC installed, go to the `ECE_`
`home/occeserver/bin` directory.
7. Start ECC:

```
./ecc
```
8. Start the charging server nodes:

```
start server
```

The subscriber lifecycle states are now enabled in ECE.