Oracle® Communications Billing and Revenue Management

Elastic Charging Engine Release Notes

12.0
F16760-03
December 2020

ORACLE

Oracle Communications Billing and Revenue Management Elastic Charging Engine Release Notes, 12.0

F16760-03

Contents

Preface

This guide includes information about Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) 12.0. Detailed information on the new features is included in the BRM 12.0 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`

1

Release Notes

This chapter describes the features introduced or enhanced in Oracle Communications Billing and Revenue Management Elastic Charging Engine (ECE) for the 12.0 (12.0.0.1) release. It also lists the documentation updates and known problems in this release.

Detailed information on the new features is included in the BRM 12.0 documentation.

ECE 12.0 Release Notes

This section provides information about ECE 12.0.

New Features

This section describes the new features introduced in ECE 12.0.

> **Note:**
>
> You can integrate ECE 12.0 (12.0.0.1) only with:
>
> - Oracle Communications Billing and Revenue Management (BRM) 12.0 Patch set 1
> - Oracle Communications Pricing Design Center (PDC) 12.0 Patch Set 1
> - Oracle Communications Offline Mediation Controller 12.0 Patch Set 1

ECE Data Can Be Persisted in Oracle Database for High Data Resilience

ECE now supports persistence of data. You can now persist the ECE cache data in an Oracle database. This feature enables you to perform the following:

- Create a backup of the ECE cache data (in-memory data) in the ECE persistence database (an Oracle database).
- Persist the data published or loaded into ECE in the persistence database.
- Persist the POID pool received from BRM in the persistence database for continued POID allocation.
- Deploy ECE without Oracle NoSQL Database and load the rated events directly into the persistence database.
- Store only limited data, such as customer data and balances, in the ECE cache and reload remaining data from the persistence database when required or at the time of ECE restart.
- Recover data from the persistence database automatically in case of a node failure or partition loss.

- Optionally load data from BRM, PDC, and files into the persistence database at the time of ECE restart.

This simplifies the data storage process, ensures high data resilience, and also reduces the downtime of the ECE system.

In addition, the pricing and configuration caches are now configured as near caches with their front local scheme being a size-limited local cache. The local cache uses a HYBRID eviction policy that constitutes LRU (Least Recently Used) and LFU (Least Frequently Used) policies. When the configured size-limit is reached, the entries that have been least recently or least frequently used are evicted. For information about the hybrid eviction policy for Coherence caches, see the Oracle Coherence documentation.

You can enable data persistence and create the schema and tablespaces for storing the ECE data during the ECE installation or upgrade. For instructions, see *BRM ECE Installation Guide*.

For more information on data persistence, see the discussion about managing persisted data in the Oracle database in *BRM System Administrator's Guide*.

Enhanced Security in ECE

ECE 12.0 includes the following security enhancements:

- The boundary system passwords (such as BRM password) are not stored in the files. The ECE Installer stores these passwords in the ECE Oracle wallet and ECE retrieves the information from this wallet. ECE does not create the **keystore.jks** file for securing boundary system passwords.
- Password restrictions are applied for all the passwords provided during installation.

For more information on Oracle wallet and managing passwords in the wallet, see the discussion about setting up and managing Elastic Charging Engine security in *BRM System Administrator's Guide*.

Documentation Updates

This section provides an overview of the documentation updates introduced in ECE 12.0.

The following changes have been made to the documentation:

- *BRM Elastic Charging Engine Implementation Guide* has been renamed as *BRM ECE Implementing Charging*.
- *BRM Elastic Charging Engine Extensions* document has been added as a chapter in *BRM ECE Implementing Charging*.
- The following guides now include information for ECE 12.0:
  - *BRM Concepts*
  - *BRM Security Guide*
  - *BRM System Administrator's Guide*

Known Problems

This section provides an overview of the known problems in ECE 12.0.

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

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*/**oceceserver/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.appconfiguration.beans.productst
ate.LifecycleState"state="101" stateName="PREACTIVE"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.productst
ate.LifecycleState"state="102" stateName="ACTIVE"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.productst
ate.LifecycleState"state="103" stateName="RECHARGE_ONLY"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.productst
ate.LifecycleState"state="104" stateName="CREDIT_EXPIRED"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.productst
ate.LifecycleState"state="105" stateName="DORMANT"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.productst
ate.LifecycleState"state="106" stateName="FRAUD_INVESTIGATED"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.productst
ate.LifecycleState"state="107" stateName="SUSPENDED"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.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.appconfiguration.beans.productst
ate.LifecycleState"state="10100" stateName="ACTIVE"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.beans.productst
ate.LifecycleState"state="10102" stateName="SUSPENDED"/>
<lifecycleStateconfig-
class="oracle.communication.brm.charging.appconfiguration.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*/**oceceserver/bin** directory.
7. Start ECC:

   ```
   ./ecc
   ```

8. Start the charging server nodes:

   ```
   start server
   ```

The subscriber lifecycle states are now enabled in ECE.

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.