Oracle® Communications Billing and Revenue Management

Calculating Taxes

Release 15.2
G35849-01
January 2026

Oracle Communications Billing and Revenue Management Calculating Taxes, Release 15.2

G35849-01

Contents

5 Creating Tax Exemptions

6 Configuring Tax Calculation Options

7 Configuring Vertex Data Manager for Vertex Q Series

8 Configuring Taxation Gateway for Vertex O Series

9 Tax Calculation Utilities

About This Content

This guide describes how to configure taxation in your Oracle Communications Billing and Revenue Management (BRM) system.

Audience

This guide is intended for those who configure BRM tax calculation.

1

About Calculating Taxes

In Oracle Communications Billing and Revenue Management (BRM), you can calculate the taxes your customers owe for purchases.

Topics in this document:

About Calculating Taxes

BRM uses tax codes to calculate the taxes for charges and discounts. For flat taxes, a tax code defines how to apply a fixed rate. For Vertex O Series, BRM maps the tax code to a Vertex O Series product or taxability driver, and Vertex returns the applicable taxes and rates by jurisdiction. When you create product offerings, you can specify the tax code to apply to charges, discounts, and chargeshares. For example, you can use one tax code for telephone handsets and another for online service subscriptions.

When configuring charges, you can specify when BRM calculates taxes. Taxes can be calculated at the following times:

- **During rating**: BRM calculates taxes when the event is rated and adds them to the customer's account balance. This approach maintains an accurate account balance throughout the accounting cycle.
- **During billing**: BRM calculates taxes during billing. Deferring tax calculation to billing can reduce rounding errors because events of the same type are taxed together. For example, BRM taxes the total usage fees rather than individual usage events.
- **During billing, but using the event time**: BRM calculates taxes during billing, using the tax rate in effect when the event occurred. This method provides the benefits of billing-time taxation while allowing tax rate changes during a billing cycle. For example, if a recurring charge's tax rate changes from 2% to 3% on May 15 and the billing cycle ends on May 30, BRM uses the 2% rate to calculate taxes for April 30 through May 14 and the 3% rate for May 15 through May 29.

BRM can use the following methods to calculate taxes:

- Use tax codes alone to apply simple flat taxes.
- Use a tax selector to apply tax codes based on attributes in the **/account**, **/service**, **/event**, or **/profile** objects.
- Use Vertex tax calculation software to apply complex taxation across multiple jurisdictions.
  - Vertex Communications Tax Q Series: Use the Vertex Data Manager to communicate with the Vertex tax database.
  - Vertex O Series: Use the Taxation Gateway, which integrates with Vertex O Series using REST APIs.

When you use Vertex to calculate taxes, you configure tax suppliers. A tax supplier is a company or corporate division responsible for collecting taxes for a transaction. Tax suppliers typically include your corporate headquarters and branch offices.

For more information about creating tax codes and tax selectors, see "Creating Tax Codes".

You can also use tax calculation opcodes to create a custom tax calculation application. See "Using Custom Tax Rates" in *BRM Opcode Guide*.

Impacts of Tax Calculation

Tax calculation affects how other BRM features work.

- You can specify whether to perform an adjustment, dispute, or settlement with or without taxes, provided the original item or event was taxable. Taxed adjustments, disputes, and settlements reverse the tax levied on the original item. See "Accounts Receivable" in *Billing Care Online Help*.
- You can record taxes as general ledger data. Create a general ledger ID (G/L ID) for each tax code. See "Setting Up Your General Ledger" in *BRM Collecting General Ledger Data*.
- When rerating taxed events, BRM calculates taxes on any deferred taxable amounts in the subsequent bill run. The rerated tax appears on the next invoice. You can configure BRM to apply deferred taxes during rerating so that corrected invoices include the deferred tax amounts.
- You can define tax exemptions for customers in Billing Care or Customer Center. For example, you can specify that an account is exempt from city taxes. You can also specify an exemption percentage (for example, 10% not taxed) or a full exemption (100%).

  Vertex Communications Tax Q Series and Vertex O Series do not support partial tax exemptions in percentage. They supports only total exemption (100%) or no exemption (0%). If you set up an account to have a partial tax exemption, BRM sends the transaction for this account to the tax package as 100% tax-exempt.

About Tax Exemptions

Tax exemptions reduce or eliminate the tax owed by customers.

You can apply tax exemptions for event-time and billing-time calculations by:

- Adding a tax exemption to an existing account.
- Using a tax exemption selector to apply exemption codes to products based on account, service, event, or profile attributes.

If you define exemptions at both the account and product levels for the same jurisdiction (for example, federal), BRM applies the account-level exemption.

If you define exemptions at the account and product levels for different jurisdictions, BRM applies both exemptions. For example, if the account has a federal exemption and the product has a state exemption, BRM applies both.

2 Calculating Taxes for Accounts Receivable Actions

In Oracle Communications Billing and Revenue Management (BRM), you can calculate taxes for adjustments, disputes, and settlements.

Topics in this document:

About Taxation for Adjustments, Disputes, and Settlements

You can configure how BRM calculates taxes for adjustments, disputes, and settlements, including whether tax reversals occur when account adjustments are created or at billing time.

You can perform adjustments, disputes, or settlements with or without taxes at the account, bill, item, and event levels. For taxable account, bill, and item adjustments, BRM uses the tax configuration (for example, **/config/ar_taxes**) to calculate the tax reversal.

If you include taxes in an A/R action:

- **If billing has already occurred**: BRM always calculates taxes when you create the adjustment, dispute, or settlement.
- **If billing has not occurred**: BRM applies the tax amount only if the item has real-time taxes. If the item has deferred taxes, BRM calculates and applies the tax amount at billing time.

To ensure tax calculations are performed correctly for adjustments, disputes, settlements, and write-offs:

- Configure a separate, tax-only item. See "Cumulative Custom Item for Taxes" in *BRM Configuring and Running Billing*.
- Ensure all events associated with an item use the same tax method (tax now or tax deferred). Nontaxable events are valid in either scenario.

About Adjustment Tax and Billing Cycles

For account, subscription service, or member service–level adjustments, BRM uses the total adjustment amount without considering the specific events included.

For bill-, item-, and event-level adjustments, BRM distinguishes between items and events that need real-time taxes and those that need deferred taxes. If some adjusted items or events are nontaxable, BRM omits them from the deferred amount transferred for taxation during billing. For partial adjustments, BRM proportions the deferred amount according to the granted portion of the adjustment.

About Dispute and Settlement Taxes

For bill- and item-level disputes and settlements, BRM applies real-time taxes when you create the dispute or settlement. When processing disputes and settlements with deferred taxes, BRM determines whether the original item has been billed.

- **If billing has not occurred**: BRM has not levied tax for the original item. No separate tax application is needed because the dispute or settlement adjusts the balance before BRM calculates taxes.
- **If billing has occurred**: BRM has already charged tax on the original item. BRM defers applying the tax reversal amount until the next billing run.

For partial disputes and settlements, BRM proportions the deferred amount by the relative size of the dispute or settlement.

Configuring Tax Calculation for Adjustments, Disputes, and Settlements

Depending on tax law, jurisdiction, and other factors, you may need to specify whether to apply a tax reversal when performing adjustments, disputes, and settlements. For example, the law may require that you reverse taxes for any adjustment you apply to a charge for a call placed outside the country but not require you to reverse taxes for adjustments to prepaid balances.

As a prerequisite, load the **/config/ar_taxes** object, which defines tax treatment parameters for each event type. BRM uses this object to enrich the adjustment, dispute, and settlement opcode flists with the tax information needed to calculate the tax reversal.

After loading the **pin_config_ar_taxes.xml** file, you have the following options:

- If you use Customer Center for accounts receivable operations, configure a default tax treatment for adjustments, disputes, and settlements. Customer Center then determines whether these activities always have a tax reversal, never have a tax reversal, or allow a CSR to decide. See "Configuring the Default Tax Treatment for Customer Center ".
- If you do not use Customer Center for adjustments, edit the CM **pin.conf** file to configure BRM to apply tax reversals at adjustment time instead of deferring until billing. See "Configuring the Default Tax Method for Account Adjustments".

  If you use Customer Center, BRM ignores the tax reversal entry in **pin.conf**. Use the Customer Center Configurator to set the tax treatment.

Before loading the tax configuration, load the tax suppliers. See "About Tax Suppliers".

The **pin_config_ar_taxes.xml** file specifies the tax supplier and tax code for each adjustment, dispute, and settlement event type.

- The tax suppliers specified in **pin_config_ar_taxes.xml** must be listed in **tax_supplier_map**.
- The tax code used in **pin_config_ar_taxes.xml** must be present in **taxcodes_map** and mapped to a package that calculates tax for the specific charge offer type. The tax code can be numeric or a descriptive string.

For information on tax suppliers, tax codes, and the associated map files, see "Creating Tax Suppliers".

The **pin_config_ar_taxes.xml** file must follow standard XML formatting rules. The following example shows a typical **pin_config_ar_taxes.xml** file.

> **Note**
>
> The **<Name>** element identifies the tax supplier.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <AccountReceivablesConfiguration>
    <TaxConfigurationList>
      <TaxConfiguration>
        <Event>/event/billing/adjustment/item</Event>
        <Name>ABC Inc - California branch</Name>
        <TaxCode>usage</TaxCode>
      </TaxConfiguration>
      <TaxConfiguration>
        <Event>/event/billing/dispute/item</Event>
        <Name>ABC Inc - California branch</Name>
        <TaxCode>usage</TaxCode>
      </TaxConfiguration>
      <TaxConfiguration>
        <Event>/event/billing/settlement/item</Event>
        <Name>ABC Inc - New Jersey branch</Name>
        <TaxCode>usage</TaxCode>
      </TaxConfiguration>
      <TaxConfiguration>
        <Event>/event/billing/adjustment/account</Event>
        <Name>ABC Inc - Illinois branch</Name>
        <TaxCode>cycle</TaxCode>
      </TaxConfiguration>
    </TaxConfigurationList>
  </AccountReceivablesConfiguration>
```

To load the tax configuration:

1. Edit the sample **pin_config_ar_taxes.xml** file in the *BRM_home***/sys/data/config** directory.

   You can rename the file and save it to a different directory.

2. Save the **pin_config_ar_taxes.xml** file.
3. Use the following command to run the **load_pin_ar_taxes** utility:

   ```
   load_pin_ar_taxes -f input_file.xml
   ```

   where *input_file.xml* is the name and path to the XML file.

4. Stop and restart the Connection Manager (CM) and, if necessary, your client application.

To verify that the **pin_config_ar_taxes.xml** file was loaded, you can display the **/config/ar_taxes** object by using the Object Browser, or use the **robj** command with the **testnap** utility. See "Reading an Object and Writing Its Contents to a File" in *BRM Developer's Guide*.

Configuring the Default Tax Treatment for Customer Center

If you use Customer Center as your CRM application, you can configure a default tax treatment for adjustments, disputes, and settlements. Customer Center then determines whether these activities always include a tax reversal, never include a tax reversal, or allow a CSR to decide. A default tax treatment helps enforce uniform tax handling for a given location and reduces CSR decision-making.

Configure the tax treatment on the **Balance** tab in Customer Center Configurator. This setting overrides any tax treatment specified in the **pin.conf** file. To set up tax treatment for adjustments, disputes, and settlements, select one of the following under **Tax treatment** on the Configurator **Balance** tab:

- **Include tax**: Always perform a tax reversal for the adjustment, dispute, or settlement.
- **Exclude tax**: Never perform a tax reversal for the adjustment, dispute, or settlement.
- **None**: Allow the CSR to choose whether to include taxes.

For information on using the Customer Center Configurator **Balance** tab and the implications of these three options, see "Balance Configurator" in *BRM Developer's Guide*.

Configuring the Default Tax Method for Account Adjustments

You can configure BRM to calculate the tax for account adjustments using one of these methods:

- **Billing-time taxation**: BRM defers tax calculation for all account adjustments until the end of the billing cycle. This is the default.
- **Adjustment-time taxation**: BRM calculates taxes when you create an account adjustment.
- **Dynamic taxation**: BRM defers tax calculation until the end of the billing cycle but uses the tax rate in effect at the time of the account adjustment (for example, an adjustment on June 15 with a June 30 bill date uses the June 15 rate). Dynamic taxation provides the benefits of billing-time taxation while allowing you to change tax rates during a billing cycle.

To configure the default tax method for account adjustments:

1. Open the Connection Manager configuration file (*BRM_home*/**sys/cm/pin.conf**).
2. Modify the **tax_now** parameter:

   ```
   -fm_ar tax_now value
   ```

   where *value* is set to one of the following:

   - **0**: Billing-time taxation. This is the default value.
   - **1**: Adjustment-time taxation.
   - **2**: Dynamic taxation.
   - **3**: Itemized taxation.

   > **Note**
   >
   > Ensure all events associated with an item use the same tax method to ensure correct tax calculations.

3. Save and close the file.

You do not need to restart the CM to enable this entry.

Configuring Itemized Taxation for A/R and Payments

You can save tax information for A/R and payments at the item level. This allows you to view the following details for each adjustment and payment:

- Settled tax amount
- Settled taxed amount
- Settled non-taxed amount

You can configure BRM to calculate itemized taxes for adjustments and payments by setting the **ItemizedTaxCalculation** business parameter to **enabled** using the **pin_bus_params**. See "pin_bus_params" in *BRM Developer's Guide* for more information about this utility.

You can view details about the taxes applied or reversed during A/R and payments at the following level:

- **Itemize**: BRM segregates tax details based on the tax code and tax jurisdiction.
- **Summarize**: BRM segregates tax details based only on the tax code.

To configure whether BRM uses itemized taxation:

1. Go to *BRM_home***/sys/data/config**.
2. Create an XML file from the **/config/business_params** object:

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

   This command creates an XML file named **bus_params_billing.xml.out** in your working directory.
3. Open the **bus_params_billing.xml.out** file.
4. Set the **ItemizedTaxCalculation** parameter:

   ```
   <ItemizedTaxCalculation>value</ItemizedTaxCalculation>
   ```

   where *value* is:
   - **enabled** to have BRM store tax information at the item level.
   - **disabled** (default) to have BRM not use itemized taxation.
5. Save and rename this file as **bus_params_billing.xml**.
6. Load the XML file into the BRM database:

   ```
   pin_bus_params  bus_params_billing.xml
   ```
7. Open the **bus_params_rating.xml.out** file.
8. In the file, set the **TaxReturnJuris** parameter:

   ```
   <TaxReturnJuris>value</TaxReturnJuris>
   ```

   where *value* is:
   - **itemize** to have BRM report **settled_tax** and **settled_taxed_amount** based on tax code and jurisdiction level.

   - **summarize** to have BRM report the consolidated amount of the **settled_tax_amount**, **settled_taxed_amount**, and **settled_non_taxed_amount** for each item.

9. Save this file as **bus_params_rating.xml**.
10. Load the XML file into the BRM database:

    ```
    pin_bus_params bus_params_rating.xml
    ```

11. Stop and restart the CM.

> **Note**
>
> AR actions are not supported for scenarios having both event time and billing time taxation in the same item.

Retrieving Itemized Tax Details

You can retrieve the itemized tax details for a customer's A/R action using one of the following:

- Billing Care. See "Accounts Receivable" in *Billing Care Online Help*.
- A custom client application calling the PCM_OP_AR_GET_ITEM_DETAIL opcode. See "Retrieving Details about a Specific A/R Item or Bill Item" in the *BRM Opcode Guide*.
- A custom client application calling the following operations in the Billing Care REST API:
  - For disputes and settlements: **Get Settlement Details for a Dispute**
  - For adjustments: **Get Details About an Adjustment**
  - For payments: **Get Details for a Payment**

  For more information, see *REST API Reference for Billing Care*.

3

Creating Tax Codes

In Oracle Communications Billing and Revenue Management (BRM), you use tax codes to assign taxes to the products and services that you sell to your customers. For example, you use tax codes to apply taxes differently for usage and physical goods charges.

Topics in this document:

About Creating Tax Codes

Tax codes define which tax to apply to a charge. For example, a telephone handset may use a different tax code than an online service subscription. The tax code specifies the tax amount or percentage to apply for each jurisdiction.

You can configure tax codes in the following ways:

- Using the PDC UI. See "Creating Tax Codes" in *PDC Online Help*.
- Using the **ImportExportPricing** utility. See "Creating Tax Codes Using ImportExportPricing".

> **Note**
>
> The process for creating tax codes for the Vertex O Series tax calculation software differs from other methods. For information, see "Creating Tax Codes for Vertex O Series".

After you create tax codes, associate them with your charges, discounts, chargeshares, and G/L IDs.

You can also assign tax codes to charges, discounts, and chargeshares using tax selectors. Tax selectors determine the tax code to apply based on account, service, event, and profile attributes. You can create tax selectors in the following ways:

- Using the PDC UI. For information, see "Creating a Tax Selector" in *PDC Online Help*.
- Using the **ImportExportPricing** utility. For information, see "Creating Tax Selectors Using ImportExportPricing".

Creating Tax Codes Using ImportExportPricing

You create tax codes in PDC and publish them to BRM and Elastic Charging Engine (ECE) using the **ImportExportPricing** utility.

To implement tax codes:

1. Define your tax codes in an XML file. You can base your tax codes on those provided in the **Sample_Tax_Code.xml** file available in the *PDC_home***/apps/Samples/Examples** directory, where *PDC_home* is the PDC installation directory. The sample file contains examples of tax codes for calculating taxes based on a custom implementation or Vertex Q Series. For more information about XML elements specific to Vertex Q Series or custom tax codes, see:
   - Creating Tax Codes for a Flat Tax or Custom Implementation
   - Creating Tax Codes for Vertex Tax Q Series
2. Import the XML file into PDC using the **ImportExportPricing** utility:

   ```
   ImportExportPricing -import -config MyTaxCodes.xml
   ```

   where *MyTaxCodes* is the name of the XML file where you defined the tax codes.

   You can modify existing tax codes using the **-ow** parameter.

   See "Importing Pricing and Setup Components" and "ImportExportPricing" in *PDC Creating Product Offerings* for more information about the utility's syntax and parameters.

   The **ImportExportPricing** utility publishes the tax codes to the real-time rating engine (RRE) and to ECE.

   > **Note**
   >
   > You cannot publish tax codes to the Batch Rating Engine (BRE).

Creating Tax Codes for a Flat Tax or Custom Implementation

You implement simple flat taxes by defining them in an XML file that you import into PDC. In the file, you enter a tax code and assign it to one or more tax rates, which are differentiated by criteria such as validity date and jurisdiction.

The following XML example shows custom tax codes:

```
<taxCodes>
      <name>CustomTaxCode</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>CustomTaxCode</code>
      <taxPackageType>CUSTOM</taxPackageType>
      <glidName>National VAS</glidName>
      <taxCodeValidityPeriods>
         <validFrom>20010101</validFrom>
         <validTo>20020120</validTo>
         <taxCodeMaps>
            <description>TRS1</description>
            <percent>4.0</percent>
            <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
            <taxJurisdictions>US</taxJurisdictions>
            <taxJurisdictions>UK</taxJurisdictions>
            <taxJurisdictions>IN</taxJurisdictions>
            <taxJurisdictions>RU</taxJurisdictions>
            <taxJurisdictions>GE</taxJurisdictions>
            <taxJurisdictions>SZ</taxJurisdictions>
            <taxRuleType>TAX</taxRuleType>
         </taxCodeMaps>
```

```
      </taxCodeValidityPeriods>
<taxCodes>
```

When the value of the **<taxPackageType>** element is **CUSTOM**, one of the following is required under the **<taxCodeMaps>** element:

- **<percent>**: If you include this element, the elements described in Table 3-1 are also required.
- **<transType_CategoryCode_Rate>**: If you include this element, no other elements are supported in the tax code map.

Table 3-1 describes the critical XML elements specific to custom tax codes.

Table 3-1 XML Elements in the Tax Code File for Custom Tax Codes

Element	Description
code	The tax code. A unique alphanumeric value that defines categories with different tax treatments.
taxPackageType	The package type. For flat taxes and custom implementations, this is always **CUSTOM**.
validFrom	The start date of the validity period for the tax rate in *yyyymmdd* format. **Note:** You can use multiple validity periods to create different rates for a single tax code, as shown in the example, but the validity periods must not overlap.
validTo	The end date of the validity period for the tax rate in *yyyymmdd* format.
description	A brief description of the tax code.
glidName	The general ledger ID associated to the tax code.
percent	The tax rate in percent. For example, **4.25** means 4.25%. For prepaid purchase events that grant negative currency balances, the corresponding tax associated with it should also be negative. For example, **-4.25**.
taxJurisdictionLevel	The jurisdiction level for which this rate is applicable. Values are **FEDERAL**, **STATE**, **COUNTY**, **CITY**, and **LOCATION**.
taxJurisdictions	The jurisdiction where the rate applies. Similar to a nexus for the corresponding jurisdiction level. For example, if the **taxJurisdictionLevel** value is **STATE**, the **taxJurisdictions** values must be state-level jurisdictions.

Table 3-1 (Cont.) XML Elements in the Tax Code File for Custom Tax Codes

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>taxRuleType</td><td>Determines how taxes will be computed. Values are:

STANDARD: Standard tax computation. Taxes are computed based on the taxable amount and are then added to the total.
TAX: "Tax on tax" computation. Taxes are computed based on previous taxable amounts and taxes, and are then added to the total.
For example, if tax1 = 10%, tax2 = 20%, and charge = 100.00, taxes are computed as follows:
tax1 = 10% @ 100.00 = 10.00
tax2 = 20% @ (100.00 + 10.00) = 22.00
INCLUSIVE: Inclusive tax computation. Taxes are already included in the total. For example, if the total is $110 and the tax is 10%, the tax amount is $10 and the charge is $100. This option can only be configured for event-time taxation, not billing-time taxation or dynamic taxation.

Note
Inclusive taxes are not supported by Oracle Communications Elastic Charging Engine (ECE).
NCS: Noncumulative standard tax computation. Taxes are computed based on the taxable amount but are not added to the total.
NCT: Noncumulative "tax on tax" computation. Taxes are computed based on the taxable amount but are not added to the total.
For example, if tax1 = 10%, tax2 = 20%, and charge = 100.00, taxes are computed as follows:
tax1 = 10% @ 100.00 = 10.00
tax2 = 20% @ 100.00 = 20.00
</td></tr>
</table>

Creating Tax Codes for Vertex Tax Q Series

You specify to calculate taxes using Vertex Tax Q Series by mapping tax codes to Vertex Tax Q Series product codes in an XML file that you import into PDC. The Vertex Tax Q Series product codes are used to apply the tax rate.

The following XML example shows the two types of Vertex Tax Q Series tax codes. The three elements under the **taxCodeMaps** element are the only ones supported for Vertex Tax Q Series tax codes.

```
<taxCodes>
      <name>Usage</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>usage</code>
      <taxPackageType>CUSTOM</taxPackageType>
      <glidName>National VAS</glidName>
      <taxCodeValidityPeriods>
         <validFrom>20010201</validFrom>
         <validTo>20010201</validTo>
         <taxCodeMaps>
            <description>USF</description>
            <percent>4.0</percent>
            <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
            <taxJurisdictions>US</taxJurisdictions>
```

```
            <taxRuleType>STANDARD</taxRuleType>
         </taxCodeMaps>
      </taxCodeValidityPeriods>
   </taxCodes>
   <taxCodes>
      <name>Cycle</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>cycle</code>
      <taxPackageType>CUSTOM</taxPackageType>
      <glidName>Cancel Fees</glidName>
      <taxCodeValidityPeriods>
         <validFrom>20010201</validFrom>
         <validTo>20300131</validTo>
         <taxCodeMaps>
            <description>Excise</description>
            <percent>3.5</percent>
            <validFrom>20010201</validFrom>
            <validTo>20300131</validTo>
            <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
            <taxJurisdictions>US</taxJurisdictions>
            <taxRuleType>STANDARD</taxRuleType>
         </taxCodeMaps>
      </taxCodeValidityPeriods>
   </taxCodes>
   <taxCodes>
      <name>Purchase</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>purchase</code>
      <taxPackageType>CUSTOM</taxPackageType>
      <glidName>Deposit Charge</glidName>
      <taxCodeValidityPeriods>
         <validFrom>20010201</validFrom>
         <validTo>20300131</validTo>
         <taxCodeMaps>
            <description>Sales</description>
            <percent>6</percent>
            <taxJurisdictionLevel>STATE</taxJurisdictionLevel>
            <taxJurisdictions>CA</taxJurisdictions>
            <taxRuleType>STANDARD</taxRuleType>
         </taxCodeMaps>
         <taxCodeMaps>
            <description>Sales</description>
            <percent>1.25</percent>
            <taxJurisdictionLevel>COUNTY</taxJurisdictionLevel>
            <taxJurisdictions>*</taxJurisdictions>
            <taxRuleType>STANDARD</taxRuleType>
         </taxCodeMaps>
         <taxCodeMaps>
            <description>Sales</description>
            <percent>1.0</percent>
            <taxJurisdictionLevel>CITY</taxJurisdictionLevel>
            <taxJurisdictions>*</taxJurisdictions>
            <taxRuleType>STANDARD</taxRuleType>
         </taxCodeMaps>
      </taxCodeValidityPeriods>
   </taxCodes>
   <taxCodes>
      <name>Toll</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
```

```
         <code>toll</code>
         <taxPackageType>CUSTOM</taxPackageType>
         <glidName>Deposit Refund</glidName>
         <taxCodeValidityPeriods>
            <validFrom>20010201</validFrom>
            <validTo>20300131</validTo>
            <taxCodeMaps>
               <description>TRS</description>
               <percent>2.0</percent>
               <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
               <taxJurisdictions>US</taxJurisdictions>
               <taxRuleType>TAX</taxRuleType>
            </taxCodeMaps>
            <taxCodeMaps>
               <description>911</description>
               <percent>1.5</percent>
               <taxJurisdictionLevel>STATE</taxJurisdictionLevel>
               <taxJurisdictions>CA</taxJurisdictions>
               <taxRuleType>TAX</taxRuleType>
            </taxCodeMaps>
            <taxCodeMaps>
               <description>B-O</description>
               <percent>2.0</percent>
               <taxJurisdictionLevel>STATE</taxJurisdictionLevel>
               <taxJurisdictions>CA</taxJurisdictions>
               <taxRuleType>TAX</taxRuleType>
            </taxCodeMaps>
            <taxCodeMaps>
               <description>Deaf</description>
               <percent>3.15</percent>
               <taxJurisdictionLevel>CITY</taxJurisdictionLevel>
               <taxJurisdictions>Cupertino</taxJurisdictions>
               <taxRuleType>TAX</taxRuleType>
            </taxCodeMaps>
         </taxCodeValidityPeriods>
      </taxCodes>
      <taxCodes>
         <name>VAT</name>
         <priceListName>Default</priceListName>
         <obsolete>false</obsolete>
         <code>VAT</code>
         <taxPackageType>CUSTOM</taxPackageType>
         <glidName>Payment received</glidName>
         <taxCodeValidityPeriods>
            <validFrom>20010201</validFrom>
            <validTo>20300131</validTo>
            <taxCodeMaps>
               <description>VAT-EU</description>
               <percent>5</percent>
               <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
               <taxJurisdictions>GB</taxJurisdictions>
               <taxJurisdictions>FR</taxJurisdictions>
               <taxRuleType>STANDARD</taxRuleType>
            </taxCodeMaps>
            <taxCodeMaps>
               <description>VAT-GB</description>
               <percent>4.5</percent>
               <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
               <taxJurisdictions>GB</taxJurisdictions>
               <taxRuleType>STANDARD</taxRuleType>
            </taxCodeMaps>
            <taxCodeMaps>
```

```
            <description>VAT-FR</description>
            <percent>4.0</percent>
            <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
            <taxJurisdictions>FR</taxJurisdictions>
            <taxRuleType>STANDARD</taxRuleType>
         </taxCodeMaps>
      </taxCodeValidityPeriods>
   </taxCodes>
   <taxCodes>
      <name>qt_usage</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>qt_usage</code>
      <taxPackageType>VERTEX_QUANTUM</taxPackageType>
      <glidName>Monthly Fees</glidName>
      <taxCodeValidityPeriods>
         <taxCodeMaps>
            <transType_CategoryCode_Rate>01</transType_CategoryCode_Rate>
            <transSubType_ServiceCode>01</transSubType_ServiceCode>
            <salesIndicator>SALE</salesIndicator>
         </taxCodeMaps>
      </taxCodeValidityPeriods>
   </taxCodes>
   <taxCodes>
      <name>qt_cycle</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>qt_cycle</code>
      <taxPackageType>VERTEX_QUANTUM</taxPackageType>
      <glidName>Monthly Fees</glidName>
      <taxCodeValidityPeriods>
         <taxCodeMaps>
            <transType_CategoryCode_Rate>01</transType_CategoryCode_Rate>
            <transSubType_ServiceCode>01</transSubType_ServiceCode>
            <salesIndicator>SALE</salesIndicator>
         </taxCodeMaps>
      </taxCodeValidityPeriods>
   </taxCodes>
   <taxCodes>
      <name>ct_direct</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>ct_direct</code>
      <taxPackageType>VERTEX_COMMTAX_21</taxPackageType>
      <glidName>Misc Adjustments</glidName>
      <taxCodeValidityPeriods>
         <taxCodeMaps>
            <transType_CategoryCode_Rate>01</transType_CategoryCode_Rate>
            <transSubType_ServiceCode>01</transSubType_ServiceCode>
            <salesIndicator>SALE</salesIndicator>
         </taxCodeMaps>
      </taxCodeValidityPeriods>
   </taxCodes>
   <taxCodes>
      <name>ct_toll</name>
      <priceListName>Default</priceListName>
      <obsolete>false</obsolete>
      <code>ct_toll</code>
      <taxPackageType>VERTEX_COMMTAX_21</taxPackageType>
      <glidName>Misc Adjustments</glidName>
      <taxCodeValidityPeriods>
         <taxCodeMaps>
```

```
                <transType_CategoryCode_Rate>04</transType_CategoryCode_Rate>
                <transSubType_ServiceCode>02</transSubType_ServiceCode>
                <salesIndicator>SALE</salesIndicator>
             </taxCodeMaps>
          </taxCodeValidityPeriods>
       </taxCodes>
       <taxCodes>
          <name>user_code</name>
          <priceListName>Default</priceListName>
          <obsolete>false</obsolete>
          <code>user_code</code>
          <taxPackageType>VERTEX_COMMTAX_21</taxPackageType>
          <glidName>National VAS</glidName>
          <taxCodeValidityPeriods>
             <taxCodeMaps>
                <percent>10.0</percent>
             </taxCodeMaps>
          </taxCodeValidityPeriods>
    </taxCodes>
```

Table 3-2 describes the important XML elements specific to Vertex Tax Q Series tax codes.

Table 3-2 XML Elements in the Tax Code File for Vertex Tax Q Series

Element	Explanation
code	The tax code. A unique alphanumeric value that defines categories with different tax treatments.
taxPackageType	The tax package you use: • **VERTEX_QUANTUM**: Vertex Communications Tax Q Series (telecommunications). • **VERTEX_COMMTAX_21**: Vertex Sales Tax Q Series (sales and use).
transType_CategoryCode_Rate	The field value is determined by the tax package you use. • For Vertex Communications Tax Q Series, this field is the category code. See the Communications Tax Q Series documentation for more information about its category codes. • For Vertex Sales Tax Q Series, this field is the transaction type. See the Sales Tax Q Series documentation for more information on transaction types.
transSubType_ServiceCode	The field value is determined by the tax package you use. • For Vertex Communications Tax Q Series, this field is the service code. Refer to the Communications Tax Q Series documentation for more information on service codes. • For Vertex Sales Tax Q Series, this field is the transaction subtype. Refer to the Sales Tax Q Series documentation for more information on transaction subtypes.
salesIndicator	For Vertex Communications Tax Q Series and Vertex Sales Tax Q Series, this is the resale flag field, which indicates if the charge offer is for sale (**SALE**) or resale (**RESALE**). The default is SALE.
glidName	The general ledger ID associated to the tax code.

See your tax package documentation for information about product codes, service codes, and service indicators.

When configuring Vertex tax calculation, you also need to configure tax suppliers. See "Creating Tax Suppliers".

Creating Tax Codes for Vertex O Series

To create tax codes for Vertex O Series:

1. In the Vertex UI, configure your product codes and taxability, using values that match your BRM tax codes.
2. In BRM, map those tax codes to Vertex O Series in the *BRM_home***/sys/cm/config_taxcodes_map.xml** file. BRM sends the tax code as the Vertex product identifier in the request, and Vertex O Series returns the applicable taxes and rates by jurisdiction.

   After you edit the file, you need to load the **config_taxcodes_map.xml** file into the BRM database.

The following XML example shows the structure of the **config_taxcodes_map.xml** file:

```
<TAXES elem="18">
   <TAX_CODE>internet</TAX_CODE>
   <TAXPKG_TAX_CODE>O</TAXPKG_TAX_CODE>
</TAXES>
```

Table 3-3 describes the important XML elements in **config_taxcodes_map.xml** XML file.

Table 3-3 XML Elements in config_taxcodes_map.xml File

Element	Description
<TAX_CODE>	The name of tax code, such as **internet**. For Vertex O Series, use the same value as the product code in the Vertex UI, and in the Taxation Gateway's **application.yaml** file.
<TAXPKG_TAX_CODE>	Specifies the tax package provider. Use **O** for Vertex O Series.

See your tax package documentation for information about product codes, service codes, and service indicators.

When configuring Vertex tax calculation, you also need to configure tax suppliers. See "Creating Tax Suppliers".

Creating Tax Selectors Using ImportExportPricing

You can create tax selectors in an XML file and publish them to PDC, BRM, and ECE using the **ImportExportPricing** utility.

To implement tax selectors using **ImportExportPricing**:

1. Define your tax selectors in an XML file. You can base them on the sample tax selectors provided in the *PDC_home***/apps/Samples/Examples/SampleTaxSelector.xml** file.

   See "Creating an XML File for Tax Selectors" for more information about the XML elements in the file.
2. Import the XML file into PDC using the **ImportExportPricing** utility:

   ```
   ImportExportPricing -import -pricing MyTaxSelectors.xml
   ```

   where *MyTaxSelectors* is the name of the XML file where you defined the tax selectors.

The **ImportExportPricing** utility publishes the tax selectors to the PDC database, real-time rating engine (RRE), and ECE.

> **Note**
>
> You cannot publish tax selectors to the Batch Rating Engine (BRE).

Creating an XML File for Tax Selectors

You can implement tax selectors by defining them in an XML file that you import into PDC. In the file, you enter a tax selector and then define the rules for assigning tax codes to charges, discounts, and chargeshares.

The following sample XML shows how to create tax selectors:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><pdc:PricingObjectsJXB
xmlns:pdc="http://xmlns.oracle.com/communications/platform/model/pricing">
<taxSelector xmlns:pdc="http://xmlns.oracle.com/communications/platform/model/pricing">
   <name>Sample_TAX_SELECTOR</name>
   <description>Sample TAX_SELECTOR</description>
   <pricingProfileName>Convergent Usage</pricingProfileName>
   <priceListName>Default</priceListName>
   <obsolete>false</obsolete>
   <stereoType>TAX_SELECTOR</stereoType>
   <productSpecName>TelcoGsmSms</productSpecName>
   <eventSpecName>EventDelayedSessionTelcoGsm</eventSpecName>
   <validityPeriod>
      <validFrom>0</validFrom>
      <rule>
         <name>Rule1</name>
         <result>
            <resultName>VAT</resultName>
         </result>
         <fieldToValueExpression>
            <operation>REGEX</operation>
            <seperator>;</seperator>
            <fieldName>EventDelayedSessionTelcoGsm.CALLED_ID</fieldName>
            <fieldValue>1800.*</fieldValue>
            <fieldKind>EVENT_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
      </rule>
      <rule>
         <name>Rule2</name>
         <result>
            <resultName>toll</resultName>
         </result>
         <fieldToValueExpression>
            <operation>REGEX</operation>
            <seperator>;</seperator>
            <fieldName>EventDelayedSessionTelcoGsm.CALLED_ID</fieldName>
            <fieldValue>.*</fieldValue>
            <fieldKind>EVENT_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
      </rule>
   </validityPeriod>
   </taxSelector></pdc:PricingObjectsJXB>
```

Table 3-4 describes the important XML elements specific to tax selectors.

Table 3-4 XML Elements in the Tax Selector File

Element	Description
name	The name of the tax selector.
description	A description of the tax selector.
pricingProfileName	The pricing profile to associate with the tax selector.
priceListName	The name of the price list to which the selector applies.
stereoType	The type of pricing object: **TAX_SELECTOR**.
productSpecName	The service type to which the tax selector applies.
eventSpecName	The event type to which the tax selector applies.
validityPeriod	Indicates when the rules in the tax selector are valid. The default effective period starts immediately and never ends.
name	The name of a rule.
resultName	The name of the tax code to apply when all criteria in the rule are met.
operation	The operator to apply, which is dependent on the field. The valid values are: **REGEX**, **EQUAL_TO**, **NOT_EQUAL_TO**, **GREATER_THAN**, **GREATER_THAN_EQUAL**, **LESS_THAN**, **LESS_THAN_EQUAL**, **IN_LIST**, and **MAPS_TO**.
seperator	The character that is used to separate the field values, such as a semicolon (**;**).
fieldName	The name of the field to evaluate.
fieldValue	The required value of the field.
fieldKind	The type of attributes to be selected for assigning the tax code. The valid values are: **EVENT_SPEC_FIELD**, **PRODUCT_SPEC_FIELD**, **CUSTOMER_SPEC_FIELD**, and **PROFILE_SPEC_FIELD**.

4 Creating Tax Suppliers

In Oracle Communications Billing and Revenue Management (BRM), the tax supplier data defines the location of your business. Vertex uses this data to determine which tax rate to apply to a product offering.

Topics in this document:

About Tax Suppliers

A tax supplier is a company or corporate division responsible for collecting taxes for a transaction. Tax suppliers typically include corporate headquarters and branch offices. Vertex uses the tax supplier to calculate taxes.

Tax supplier data can include:

- The name of the tax supplier.
- The address to use as the ship-from locale.
- The tax nexus state. You can have a tax nexus in any state where your company has a substantial presence, as defined by federal tax laws.
- The VAT certificate number.

Tax software typically calculates taxes by using four locales:

- **Shipped-to**: The address of the customer who made the purchase. BRM obtains the shipped-to address from the customer's account. For telecommunications taxation, this is the termination number.

  > **Note**
  >
  > You can configure BRM to use a customer’s real-time location as the shipped-to address. This option applies to third-party client applications only. See "Overriding Shipped-To with the Real-Time Location".

- **Shipped-from**: The address of the company supplying the product. For telecommunications taxation, this is the origination number.
- **Bill origin**: The location from which the product was shipped (also called *point of origin*); for example, a warehouse. This is not used for telecommunications taxation.

- **Bill approval**: The location where the order for the product was taken (also called *point of acceptance*); for example, the location of a customer service representative who creates accounts. For telecommunications taxation, this is the charge-to number.

For sales and use taxes, BRM does not differentiate between ship-from, bill origin, and bill approval. Therefore, when providing the tax software with data, BRM uses the same value for all three locales. This value is derived from the tax supplier's address. For telecommunications taxation, the origination and termination numbers are derived from call detail records (CDRs).

> **Note**
>
> For telecommunications taxation, the shipped-to, bill origin, and bill approval locales are not used and are stored in BRM with a NULL value.

If you are using the Vertex Tax Q Series software, you specify how to supply tax supplier data to the software in one of the following ways:

- Provide the tax supplier data with each transaction that you send to the tax software. For example, each transaction might include the ship-from locale based on the tax supplier address and the applicable VAT certificate. See "Providing the Tax Supplier Data with Each Transaction" for more information.
- Allow the tax software to determine the correct tax supplier based on the transaction data plus the company and business locations you give the tax software. You provide information to the tax software by running the tax software's toolkits. The tax software then determines which ship-from locale to use, if there is a tax nexus presence, and which VAT certificate to use. See "Allowing Your Sales and Use Tax Software to Determine Tax Supplier Data" for more information.

If you are using the Vertex O Series tax software, see "Providing the Tax Supplier Data for Vertex O Series".

You can also:

- Configure a default ship-from address in case BRM cannot find any tax supplier. See "Defining a Default Ship-From Locale".
- Override the shipped-to locale with your customers' real-time location when calculating taxes. See "Overriding Shipped-To with the Real-Time Location".

Providing the Tax Supplier Data with Each Transaction

> **Note**
>
> This applies only to Vertex Tax Q Series. If you use Vertex O Series, see "Providing the Tax Supplier Data for Vertex O Series".

If your business uses only one tax supplier, define a default tax supplier for all charge offers. You do not need to enter any tax supplier information when creating charge offers.

You can specify multiple tax suppliers. Use PDC or Pricing Center to assign each charge offer to a tax supplier. When an event is rated, the charge offer’s tax supplier is sent to the tax software. For example, if you have different charge offers for different states, you can indicate the state where the offer was sold by selecting the tax supplier for that state.

To use this method, edit the *BRM_home*/**sys/data/config/pin_tax_supplier.xml** file. You load tax suppliers into the database by running the **load_tax_supplier** utility.

You cannot add or change tax suppliers individually. Each time you run **load_tax_supplier**, you overwrite existing data with the entire contents of **pin_tax_supplier.xml**. To modify or remove tax suppliers, edit the file before loading it.

By default, the **pin_tax_supplier.xml** file is located in *BRM_home*/**sys/data/config**.

In **pin_tax_supplier.xml**, add a **<TaxSupplierElement>** child element to the **<TaxSupplierConfiguration>** parent element for each tax supplier.

A **<TaxSupplierElement>** looks like this:

```
<TaxSupplierElement>
   <Name>supplier_name</Name>
   <Description>supplier_description</Description>
   <Address>supplier_address</Address>
   <NexusInfo>nexus_values</NexusInfo>
   <RegulatedFlag>flag_value</RegulatedFlag>
   <DefaultFlag>flag_value</DefaultFlag>
   <VATInfo>
      <CanonCountry>country_code</CanonCountry>
      <VATCertificate>certificate_number</VATCertificate>
   </VATInfo>
</TaxSupplierElement>
```

Table 4-1 describes the **pin_tax_supplier.xml** file.

Table 4-1 pin_tax_supplier.xml File

Entry	Description
Name	Enter the tax supplier name. This name is the company ID in the **tax_supplier_map** file. For Vertex, this name is the company ID in the TDM or company file. If you use the TDM, the entry will be a four-character code. The name must match exactly.
Description	Enter a description for the tax supplier.
Address	Enter the address of the tax supplier. Use the following format: `city;state;zip_code;country`
NexusInfo	Enter a semicolon-delimited list of states where the tax supplier has a tax nexus. If you use Vertex taxation software, enter an asterisk (*****) to allow Vertex to determine the tax nexus. See "Configuring Taxation Gateway for Vertex O Series" for more information.
RegulatedFlag	For Communications Tax Q Series, enter whether the utility is regulated (**1**) or unregulated (**0**).
DefaultFlag	Enter **1** for the default tax supplier, and enter **0** for all other tax suppliers. If you have multiple tax suppliers, only one can be the default.
CanonCountry	Enter the country name. See the Vertex documentation for a list of valid values.

Table 4-1 (Cont.) pin_tax_supplier.xml File

Entry	Description
VATCertificate	Enter the VAT certificate number associated with the tax supplier. BRM currently does not use this entry. This entry can be used in a custom implementation of tax calculation.

The following **pin_tax_supplier.xml** file contains two tax suppliers; the first is the default tax supplier.

```
<TaxSupplierConfiguration>
  <TaxSupplierElement>
     <Name>TS_1</Name>
     <Description>Tax Supplier 1</Description>
     <Address>Cupertino;CA;95014;US</Address>
     <NexusInfo>*</NexusInfo>
     <RegulatedFlag>0</RegulatedFlag>
     <DefaultFlag>1</DefaultFlag>
     <VATInfo>
        <CanonCountry>US</CanonCountry>
        <VATCertificate>vat_cert_US</VATCertificate>
     </VATInfo>
     <VATInfo>
        <CanonCountry>UK</CanonCountry>
        <VATCertificate>vat_cert_UK</VATCertificate>
     </VATInfo>
  </TaxSupplierElement>
  <TaxSupplierElement>
     <Name>TS_2</Name>
     <Description>Tax Supplier 2</Description>
     <Address>New York;NY;10013;US</Address>
     <NexusInfo>*</NexusInfo>
     <RegulatedFlag>0</RegulatedFlag>
     <DefaultFlag>0</DefaultFlag>
     <VATInfo>
        <CanonCountry>US</CanonCountry>
        <VATCertificate>vat_cert_US</VATCertificate>
     </VATInfo>
     <VATInfo>
        <CanonCountry>UK</CanonCountry>
        <VATCertificate>vat_cert_UK</VATCertificate>
     </VATInfo>
  </TaxSupplierElement>
</TaxSupplierConfiguration>
```

To load tax suppliers:

1. Enter the following command to load the tax supplier data:

   ```
   load_tax_supplier pin_tax_supplier.xml
   ```

   > **Note**
   >
   > If you are not in the same directory as **pin_tax_supplier.xml**, include the full path to the file.

2. Stop and restart the CM.

Allowing Your Sales and Use Tax Software to Determine Tax Supplier Data

> **Note**
>
> This applies only to Vertex Tax Q Series. If you use Vertex O Series, see "Providing the Tax Supplier Data for Vertex O Series".

If your business has customers in multiple countries or operates under other conditions where complex tax laws are involved, allow your sales and use tax software to determine the correct tax supplier. The trade-off is that you must include every BRM charge offer in the **tax_supplier_map** file and update it when you add or delete charge offers.

The **tax_supplier_map** file lets your tax package determine the correct tax supplier company ID and business location, based on the charge offer and the ship-to locale.

For Sales Tax Q Series, the **tax_supplier_map** file also maps BRM tax supplier names to your company IDs stored in the company profile or Sales Tax Q Series TDM file. BRM uses this information to maintain database integrity.

The following example shows part of a **tax_supplier_map** file:

```
Product     ShipTo     Company ID     Business loc  Ship From              Reg
______________________________________________________________________________
Item_One : ;;CA;US : Tax supplier1 : loc_1       : Cupertino;CA;95014;US :  1
Sub_One  : ;;;US   : Tax supplier1 : loc_1       : Cupertino;CA;95014;US :  1
Sub_One  : ;;;FR   : Tax supplier2 : loc_2       : Paris;;;FR            :  0
```

Table 4-2 lists the entries in the **tax_supplier_map** file.

Table 4-2   Entries in tax_supplier_map File

Entry	Description
Product	The name of the charge offer for which taxes are calculated. See "Creating Pricing Setup Components" in *PDC Creating Product Offerings*.
ShipTo	The address the product is shipped to (your customer's location). Enter the address in this format: `city;state;ZIP;country` You can leave portions of this string blank, but you must include the semicolons for each address element. Examples: • ;;;US • ;CA;95003;US In most cases, you do not need to enter the city. See the Vertex documentation for correct state and country entries.

Table 4-2 (Cont.) Entries in tax_supplier_map File

Entry	Description
Company ID	The company ID specified in the Vertex TDM (Tax Decision Maker) or company file. This name is also stored in the PIN_FLD_NAME field when you define tax suppliers. This entry must match the name in the TDM or company file. If you use the TDM, the entry will be a four-character code.
Business Loc	The business location specified in the Vertex TDM file or company file. This value indicates which location should be considered when taxes are calculated.
ShipFrom	The address the product is shipped from. Enter the address in this format: *`city`*`;`*`state`*`;`*`ZIP`*`;`*`country`* You can leave portions of this string blank, but you must include the semicolons for each address element. Examples: • ;;;US • ;CA;95003;US The ship-from address you define in the **tax_supplier_map** file can differ from the address used when you specified the default tax supplier. In this case, your tax package obtains the address from the **tax_supplier_map** file. This enables you to create multiple ship-from addresses for a single tax supplier. See the Vertex documentation for correct state and country entries.
Regulated	For telecommunications tax only, this flag identifies the utility or company that is billing as regulated (**0**) or unregulated (**1**). Some taxes apply to regulated or unregulated utilities only; this flag determines which types of taxes are used.

To specify the location of the **tax_supplier_map** file:

1. Open the CM configuration file (*BRM_home*/**sys/cm/pin.conf**).
2. Add the **tax_supplier_map** entry. For example:
   ```
   - fm_rate   tax_supplier_map  BRM_home/sys/cm/tax_supplier_map
   ```
3. Save and close the file.
4. Stop and restart the CM.

Specifying Divisions for Tax Suppliers

> **Note**
>
> This applies only to Vertex Tax Q Series. If you use Vertex O Series, see "Providing the Tax Supplier Data for Vertex O Series."

Some companies operate divisions with locations that can impact the tax calculation. Use the **Business Loc** column in the **tax_supplier_map** file to specify divisions for tax suppliers. The data in this column is mapped to the **Division** field in Sales Tax Q Series. This value must match the Division value stored in the Sales Tax Q Series TDM database.

For example, suppose the tax supplier Acme has two divisions, one in California and one in Illinois. You can configure the **tax_supplier_map** file to assign customers to the appropriate location. For example:

```
# Product    ShipTo    Company Id  Business Loc   ShipFrom              Regulated
# -------    ------    ----------  ------------   --------              ---------
electrical : ;;CA;US : Acme      : West         : Cupertino;CA;95014;US : 1
electrical : ;;TX;US : Acme      : Central      : Oak Brook;IL;60523;US : 1
```

With this configuration, for customers in California, BRM sends **West** as the division to Sales Tax Q Series. The tax supplier's address will be Cupertino, CA. Similarly, customers in Texas will be assigned to the **Central** division with the Oak Brook, IL address.

Providing the Tax Supplier Data for Vertex O Series

> **Note**
>
> This only applies to Vertex O Series.

BRM sends tax supplier and location details with each tax request to Vertex O Series by using the Taxation Gateway. Each request includes the tax supplier's name and the customer's destination, along with line items and prices. Vertex O Series returns jurisdiction-level results, including effective rates, calculated taxes, and document totals used for billing. To support this flow, BRM maps tax codes to Vertex O Series by using the **config_taxcodes_map.xml** file.

You configure the Vertex O Series provider, endpoints, and mapper specifications in the BRM **application.yaml** file, and you define how each tax code is treated by Vertex.

Defining a Default Ship-From Locale

Use this option to provide a ship-from locale if a tax supplier cannot be found. You must define at least one tax supplier.

To define a default ship-from locale:

1. Open the CM configuration file (*BRM_home*/**sys/cm/pin.conf**).
2. Edit the **provider_loc** entry. For example:

   ```
   - fm_rate_pol provider_loc Middletown, CA 95222 USA
   ```

3. Save and close the file.

The new value becomes effective immediately; subsequent tax calculations use the updated address. A CM restart is not required.

Overriding Shipped-To with the Real-Time Location

This option overrides the shipped-to locale with the customer’s real-time location when calculating taxes. Using a customer’s real-time location is supported only for bundle purchases, charge offer purchases, and prepaid top-ups through a third-party client application.

Configure your third-party client application to call these opcodes with the PIN_FLD_LOCALE input flist field set to "*City*;*State*;*Zipcode*;*Country*":

- PCM_OP_SUBSCRIPTION_PURCHASE_DEAL
- PCM_OP_SUBSCRIPTION_PURCHASE_PRODUCT
- PCM_OP_PYMT_TOPUP

Alternatively, configure real-time locations through zone maps.

For more information, see "Overriding Customer Tax Locale for Bundle Purchases" in *BRM Opcode Guide*.

5

Creating Tax Exemptions

In Oracle Communications Billing and Revenue Management (BRM), you use tax exemptions to remove tax liability (either entirely or partially) from a charge. For example, you use tax exemptions to exempt a customer account from all tax liability.

Topics in this document:

- About Creating Tax Exemptions for an Account
- About Creating Tax Exemptions for Products
- Creating Product-Level Tax Exemption Codes Using ImportExportPricing
- Creating Tax Exemption Selectors Using ImportExportPricing

See "About Tax Exemptions" for more information.

About Creating Tax Exemptions for an Account

You can apply account-level tax exemptions after an account has been created by using the following:

- Billing Care. For information, see "Adding Tax Exemptions" in *Billing Care Online Help*.
- A custom client application calling the PCM_OP_CUST_SET_TAXINFO opcode. For information, see "Adding Tax Information to Accounts" in *BRM Opcode Guide*.
- A custom client application calling the Update an Account's Tax Settings endpoint in the Billing Care REST API. For information, see *REST API Reference for Billing Care*.

> **Note**
>
> Because tax exemptions can be applied only after creating an account, tax exemptions do not apply to charges incurred when you create an account.

About Creating Tax Exemptions for Products

You configure product-level tax exemptions by first creating tax exemption codes, which define the exemption rate for each tax jurisdiction level. You then create a tax exemption selector that assigns tax exemption codes to product purchases based on one or more account, service, event, and profile attributes.

To create product-level tax exemptions:

1. Create one or more tax exemption codes using the following:
   - The Pricing Design Center (PDC) UI. See "Creating Tax Exemption Codes" in *PDC Online Help*.
   - The **ImportExportPricing** utility. See "Creating Product-Level Tax Exemption Codes Using ImportExportPricing".

2. Create one or more tax exemption selectors using the following:
   - The PDC UI. See "Creating a Tax Exemption Selector" in *PDC Online Help*.
   - The **ImportExportPricing** utility. See "Creating Tax Exemption Selectors Using ImportExportPricing".

After configuring your tax exemptions, you can associate tax exemption selectors with your charge offers.

Creating Product-Level Tax Exemption Codes Using ImportExportPricing

You can create tax exemption codes in an XML file that you publish to PDC, BRM, and Elastic Charging Engine (ECE) using the **ImportExportPricing** utility.

To create tax exemption codes using **ImportExportPricing**:

1. Define your tax exemption codes in an XML file. You can base your tax exemption codes on those provided in the **Sample_Tax_Exemption_Code.xml** file available in the *PDC_home***/apps/Samples/Examples** directory, where *PDC_home* is the directory in which the PDC software is installed.

   The file contains examples of tax exemption codes for different jurisdictions. See "Creating an XML File for Tax Exemption Codes" for more information about the XML elements in the file.
2. Import the XML file into PDC using the **ImportExportPricing** utility:

   ```
   ImportExportPricing -import -config MyTaxExemptions.xml [-ow]
   ```

   where *MyTaxExemptions* is the name of the XML file where you defined the tax exemption codes.

   To modify existing tax exemption codes, include the **-ow** parameter.

   See "Importing Pricing and Setup Components" and "ImportExportPricing" in *PDC Creating Product Offerings* for more information about the utility's syntax and parameters.

The **ImportExportPricing** utility publishes the tax exemption codes to the PDC database, the real-time rating engine (RRE), and ECE.

> **Note**
>
> You cannot publish tax exemption codes to the Batch Rating Engine (BRE).

Creating an XML File for Tax Exemption Codes

You can implement tax exemption codes by defining them in an XML file that you import into PDC. In the file, you enter a tax exemption code and then assign criteria such as a validity date and jurisdiction to it.

The following XML example shows tax exemption codes:

```
<?xml version="1.0" encoding="UTF-8"?>
<cim:ConfigObjects xmlns:cim="http://xmlns.oracle.com/communications/platform/model/
Config">
    <taxExemptionCodes>
        <name>US Tax Exemption</name>
```

```
        <description>Tax Exemption for US</description>
        <priceListName>Default</priceListName>
        <code>USATaxExemptionCode</code>
        <taxExemptionCodeMaps>
            <validFrom>20010101</validFrom>
            <validTo>20321231</validTo>
            <percent>0.375</percent>
            <jurisdiction>FEDERAL</jurisdiction>
        </taxExemptionCodeMaps>
        <taxExemptionCodeMaps>
            <validFrom>20100101</validFrom>
            <validTo>20301231</validTo>
            <percent>0.75</percent>
            <jurisdiction>LOCATION</jurisdiction>
        </taxExemptionCodeMaps>
        <taxExemptionCodeMaps>
            <percent>0.5</percent>
            <jurisdiction>STATE</jurisdiction>
        </taxExemptionCodeMaps>
        <taxExemptionCodeMaps>
            <validFrom>20080601</validFrom>
            <validTo>20280531</validTo>
            <percent>0.25</percent>
            <jurisdiction>COUNTY</jurisdiction>
        </taxExemptionCodeMaps>
    </taxExemptionCodes>
    <taxExemptionCodes>
        <name>India Tax Exemption</name>
        <description>Tax Exemption for India</description>
        <priceListName>Default</priceListName>
        <code>IndiaTaxExemptionCode</code>
        <taxExemptionCodeMaps>
            <validFrom>20080101</validFrom>
            <percent>0.675</percent>
            <jurisdiction>FEDERAL</jurisdiction>
        </taxExemptionCodeMaps>
        <taxExemptionCodeMaps>
            <percent>0.725</percent>
            <jurisdiction>LOCATION</jurisdiction>
        </taxExemptionCodeMaps>
        <taxExemptionCodeMaps>
            <validFrom>20200101</validFrom>
            <validTo>20301231</validTo>
            <percent>0.25</percent>
            <jurisdiction>STATE</jurisdiction>
        </taxExemptionCodeMaps>
    </taxExemptionCodes>
</cim:ConfigObjects>
```

Table 5-1 describes the important XML elements specific to tax exemption codes.

Table 5-1 XML Elements in the Tax Exemption File

Element	Description
name	The name of the tax exemption code.
description	A description of the tax exemption code.
priceListName	The name of the price list to which the exemption applies.
code	The name of the relevant tax exemption code. It is a unique alphanumeric value that defines categories with different tax treatments.

Table 5-1 (Cont.) XML Elements in the Tax Exemption File

Element	Description
validFrom	The start date of the validity period for the tax exemption code in *yyyymmdd* format. **Note:** You can use multiple validity periods to create different rates for a single tax exemption code, but the validity periods must not overlap.
validTo	The end date of the validity period for the tax exemption code in *yyyymmdd* format.
percent	The tax exemption amount in decimal format. For example, **0.5** means 50%.
jurisdiction	The jurisdiction level for which this rate is applicable. Values are **FEDERAL**, **STATE**, **COUNTY**, **CITY**, and **LOCATION**.

Creating Tax Exemption Selectors Using ImportExportPricing

You can create tax exemption selectors in an XML file and publish them to PDC, BRM, and ECE using the **ImportExportPricing** utility.

To implement tax exemption selectors using **ImportExportPricing**:

1. Define your tax exemption selectors in an XML file. You can base them on the sample tax exemption selectors provided in the *PDC_home*/**apps/Samples/Examples/SampleTaxExemptionSelector.xml** file.

   See "Creating an XML File for Tax Exemption Selectors" for more information about the XML elements in the file.

2. Import the XML file into PDC using the **ImportExportPricing** utility:

   ```
   ImportExportPricing -import -pricing MyTaxExemptionSelectors.xml
   ```

   where *MyTaxExemptionSelectors* is the name of the XML file where you defined the tax exemption selectors.

The **ImportExportPricing** utility publishes the tax exemption selectors to the PDC database, real-time rating engine (RRE), and ECE.

> **Note**
>
> You cannot publish tax exemption selectors to the Batch Rating Engine (BRE).

Creating an XML File for Tax Exemption Selectors

You can implement tax exemption selectors by defining them in an XML file that you import into PDC. In the file, you enter a tax exemption selector and then define the rules for assigning tax exemption codes to product purchase events.

The following sample XML shows how to create tax exemption selectors:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<pdc:PricingObjectsJXB xmlns:pdc="http://xmlns.oracle.com/communications/platform/model/
pricing">
<taxExemptionSelector xmlns:pdc="http://xmlns.oracle.com/communications/platform/model/
pricing">
```

```
   <name>Sample TaxExemptionSelector</name>
   <description>Sample TaxExemptionSelector</description>
   <pricingProfileName>Product Offering</pricingProfileName>
   <priceListName>Default</priceListName>
   <obsolete>false</obsolete>
   <stereoType>TAX_EXEMPTION_SELECTOR</stereoType>
   <productSpecName>TelcoGprs</productSpecName>
   <eventSpecName>EventSessionTelcoGprs</eventSpecName>
   <customerSpecName>Account</customerSpecName>
   <validityPeriod>
      <validFrom>0</validFrom>
      <rule>
         <name>TES_rule1</name>
         <result>
            <resultName>USATaxExemptionCode</resultName>
         </result>
         <fieldToValueExpression>
            <operation>EQUAL_TO</operation>
            <seperator>;</seperator>
            <fieldName>TelcoGprs.GPRS_INFO.ROUTING_AREA</fieldName>
            <fieldValue>National</fieldValue>
            <fieldKind>PRODUCT_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
         <fieldToValueExpression>
            <operation>EQUAL_TO</operation>
            <seperator>;</seperator>
            <fieldName>EventSessionTelcoGprs.TELCO_INFO.PRIMARY_MSID</fieldName>
            <fieldValue>PMSID10248112193</fieldValue>
            <fieldKind>EVENT_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
         <fieldToValueExpression>
            <operation>IN_LIST</operation>
            <seperator>;</seperator>

<fieldName>EventBillingProductFeeCycleCycle_forward_monthly.CYCLE_INFO.CYCLE_START_T</
fieldName>
            <fieldValue>20100101</fieldValue>
            <fieldKind>EVENT_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
         <fieldToValueExpression>
            <operation>IN_LIST</operation>
            <seperator>;</seperator>

<fieldName>EventBillingProductFeeCycleCycle_forward_monthly.CYCLE_INFO.CYCLE_END_T</
fieldName>
            <fieldValue>20301231</fieldValue>
            <fieldKind>EVENT_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
         <fieldToValueExpression>
            <operation>EQUAL_TO</operation>
            <seperator>;</seperator>
            <fieldName>AnyEvent.USAGE_TYPE</fieldName>
            <fieldValue>Downlink</fieldValue>
            <fieldKind>EVENT_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
         <fieldToValueExpression>
            <operation>EQUAL_TO</operation>
            <seperator>;</seperator>
            <fieldName>Account.ACCOUNT_NO</fieldName>
            <fieldValue>ID12221521</fieldValue>
            <fieldKind>CUSTOMER_SPEC_FIELD</fieldKind>
         </fieldToValueExpression>
```

```
        </rule>
        <rule>
           <name>TES_rule2</name>
           <result>
              <resultName>IndiaTaxExemptionCode</resultName>
           </result>
           <fieldToValueExpression>
              <operation>EQUAL_TO</operation>
              <seperator>;</seperator>
              <fieldName>TelcoGprs.GPRS_INFO.ROUTING_AREA</fieldName>
              <fieldValue>SouthWest</fieldValue>
              <fieldKind>PRODUCT_SPEC_FIELD</fieldKind>
           </fieldToValueExpression>
           <fieldToValueExpression>
              <operation>EQUAL_TO</operation>
              <seperator>;</seperator>
              <fieldName>EventSessionTelcoGprs.TELCO_INFO.PRIMARY_MSID</fieldName>
              <fieldValue>PMSID10265018152</fieldValue>
              <fieldKind>EVENT_SPEC_FIELD</fieldKind>
           </fieldToValueExpression>
           <fieldToValueExpression>
              <operation>IN_LIST</operation>
              <seperator>;</seperator>

<fieldName>EventBillingProductFeeCycleCycle_forward_monthly.CYCLE_INFO.CYCLE_START_T</
fieldName>
              <fieldValue>20200101</fieldValue>
              <fieldKind>EVENT_SPEC_FIELD</fieldKind>
           </fieldToValueExpression>
           <fieldToValueExpression>
              <operation>IN_LIST</operation>
              <seperator>;</seperator>

<fieldName>EventBillingProductFeeCycleCycle_forward_monthly.CYCLE_INFO.CYCLE_END_T</
fieldName>
              <fieldValue>20321231</fieldValue>
              <fieldKind>EVENT_SPEC_FIELD</fieldKind>
           </fieldToValueExpression>
           <fieldToValueExpression>
              <operation>EQUAL_TO</operation>
              <seperator>;</seperator>
              <fieldName>AnyEvent.USAGE_TYPE</fieldName>
              <fieldValue>Uplink</fieldValue>
              <fieldKind>EVENT_SPEC_FIELD</fieldKind>
           </fieldToValueExpression>
           <fieldToValueExpression>
              <operation>EQUAL_TO</operation>
              <seperator>;</seperator>
              <fieldName>Account.ACCOUNT_NO</fieldName>
              <fieldValue>ID22220531</fieldValue>
              <fieldKind>CUSTOMER_SPEC_FIELD</fieldKind>
           </fieldToValueExpression>
        </rule>
     </validityPeriod>
</taxExemptionSelector>
</pdc:PricingObjectsJXB>
```

Table 5-2 describes the important XML elements specific to tax exemption selectors.

Table 5-2 XML Elements in the Tax Exemption Selector File

Element	Description
name	The name of the tax exemption selector.
description	A description of the tax exemption selector.
pricingProfileName	The pricing profile to associate with the tax exemption selector.
priceListName	The name of the price list to which the selector applies.
stereoType	The type of pricing object: **TAX_EXEMPTION_SELECTOR**.
productSpecName	The service type to which the tax exemption selector applies.
eventSpecName	The event type to which the tax exemption selector applies.
customerSpecName	The tag to use when the service type is **Account**.
validityPeriod	Indicates when the rules in the tax exemption selector are valid. The default effective period starts immediately and never ends.
name	The name of a rule.
resultName	The name of the tax exemption code to apply when all criteria in the rule are met.
operation	The operator to apply. Values are **EQUAL_TO** and **IN_LIST**.
seperator	The character that is used to separate the field values, such as a semicolon (**;**).
fieldName	The name of the field to evaluate.
fieldValue	The required value of the field.
fieldKind	The type of attributes to be selected for assigning the tax exemption code. The valid values are: **EVENT_SPEC_FIELD**, **PRODUCT_SPEC_FIELD**, **CUSTOMER_SPEC_FIELD**, and **PROFILE_SPEC_FIELD**.

6

Configuring Tax Calculation Options

In Oracle Communications Billing and Revenue Management (BRM), you can configure global tax calculation settings.

Topics in this document:

- Configuring Tax Calculation for Account Groups
- Validating Customer Addresses
- Reporting Zero Tax Amounts
- Reporting Taxes of Zero Percent
- Enabling and Disabling Taxation Globally
- Configuring Taxation in Pricing Center
- Configuring Taxation in loadpricelist
- Configuring Tax Calculation in ECE

Configuring Tax Calculation for Account Groups

In an account group, parent accounts always pay taxes for child accounts with nonpaying bill units (**/billinfo** objects). You can calculate taxes for account groups in two ways. Taxes can be calculated for each nonpaying child bill unit and listed as separate items on the parent bill, or consolidated into a single item for the parent and nonpaying bill units.

Specify how BRM calculates taxes for account groups by using the **CycleTaxInterval** business parameter.

> **Note**
>
> Regardless of the option you use, Bill Now always rolls activities for each nonpaying bill unit (**/billinfo**) into the parent bill unit and calculates taxes for the parent only. The parent's single tax item includes taxes from both the parent and nonpaying bill units.

To specify how to calculate taxes for account groups:

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

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

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

   ```
   <CycleTaxInterval>value</CycleTaxInterval>
   ```

where *value* is one of the following:

- **billing**: Consolidates taxes into a single item for the paying parent and nonpaying child bill units. Total tax is calculated at the parent level using the aggregated total due.
- **accounting**: Calculates taxes per nonpaying child bill unit; the total is the sum of taxes calculated for each bill unit.

The default value is **accounting**.

> **Caution**
>
> BRM uses the XML in this file to overwrite the existing instance of the **/config/business_params** object. If you delete or modify any other parameters in the file, these changes affect the associated aspects of the BRM configuration.

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

   ```
   pin_bus_params bus_params_billing.xml
   ```

   You should run this command from the *BRM_home***/sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.

   For general instructions on using **testnap**, see "Using the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.
8. Stop and restart the Connection Manager (CM).

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

Validating Customer Addresses

For U.S. customers, the tax rate is partially determined by the customer's ZIP code. If the ZIP code is invalid or belongs to a different state, the tax software calculates the taxes as **0**.

To avoid this problem, you can configure BRM to validate the customer's state and ZIP code when the customer creates an account.

Some third-party tax packages provide address validation. You can enable address validation by setting the **tax_valid** entry in the CM **pin.conf** file to the appropriate package value. The validation performed depends on the software you specify:

- Vertex Sales Tax Q Series validates city, state, and ZIP codes.
- Vertex Communications Tax Q Series does not support address validation. If you configure BRM to use Communications Tax Q Series for validation, it always returns a positive result.
- Vertex O Series validates city, state, and ZIP codes.

To specify a tax package to validate addresses:

1. Open the CM configuration file (*BRM_home*/**sys/cm/pin.conf**).
2. Set the value of the **tax_valid** entry to one of the values in Table 6-1.

Table 6-1 tax_valid Values

Value	Description
0	BRM does not check the ZIP code. This is the default value.
3	BRM uses Vertex Sales Tax Q Series to validate the ZIP code. (The Vertex DM must be running.)
4	No effect. BRM uses Communications Tax Q Series to validate the ZIP code, but Communications Tax Q Series does not perform any validation. A positive result is always returned.
6	BRM uses Vertex O Series to validate the ZIP code. (The Taxation Gateway server must be running.)

For example:

```
- fm_cust_pol tax_valid 3
```

3. Save and close the file.
4. Stop and restart the CM.

Reporting Zero Tax Amounts

By default, when taxation returns a zero amount, BRM does not report the tax amount and does not create the TAX_JURISDICTIONS array. However, legal requirements in some jurisdictions demand that zero tax amounts be reported.

To configure BRM to report zero taxes:

1. In the CM configuration file (*BRM_home*/**sys/cm/pin.conf**), set the **include_zero_tax** entry to **1**:

```
- fm_rate include_zero_tax 1
```

2. Stop and restart the CM.

To itemize taxes so zero-tax lines appear by jurisdiction:

1. In the CM **pin.conf** file, set the **tax_return_juris** entry to **itemize**:

```
- fm_rate tax_return_juris itemize
```

2. Stop and restart the CM.

Reporting Taxes of Zero Percent

When a charge's tax rate is zero percent, BRM reports a zero tax percentage and sets PIN_FLD_BAL_IMPACTS to **0** by default. To configure this feature, run the **pin_bus_params** utility to change the **ExcludePercentForZeroTaxAmt** business parameter. For more information about this utility, see "pin_bus_params" in *BRM Developer's Guide*.

> **Note**
>
> This business parameter applies to flat taxes and custom tax implementations only. It does not apply when Vertex is used for tax calculation.

To configure whether BRM reports a zero tax percentage amount:

1. Go to *BRM_home*/**sys/data/config**.
2. Create an XML file from the **/config/business_params** object:

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

3. In the file, set the **ExcludePercentForZeroTaxAmt** parameter:

   ```
   <ExcludePercentForZeroTaxAmt>value</ExcludePercentForZeroTaxAmt>
   ```

   where *value* is:

   - **enabled**: BRM reports a zero tax percentage amount.
   - **disabled**: BRM does not report zero tax percentage amounts.
4. Save this file as **bus_params_billing.xml**.
5. Load the XML file into the BRM database:

   ```
   pin_bus_params  bus_params_billing.xml
   ```

6. Stop and restart the CM.

Enabling and Disabling Taxation Globally

You can enable or disable BRM taxation globally by using the **taxation_switch** entry in the *BRM_home*/**sys/cm/pin.conf** file. You can choose to enable real-time tax calculation, deferred tax calculation, or both (the default setting). You can also disable tax calculation entirely.

> **Note**
>
> Deferred taxation procedures are triggered even if taxation features are not configured. Disabling taxation can improve performance. You can toggle taxation for testing when BRM is configured for taxation.

To enable or disable taxation globally:

1. Open the CM **pin.conf** file.
2. Set **taxation_switch** to one of the following:
   - **0**: Disable tax calculation entirely.
   - **1**: Enable only real-time tax calculation.
   - **2**: Enable only deferred (cycle-time) tax calculation.
   - **3**: (Default) Enable both real-time and deferred tax calculation.

   For example, this specifies to enable real-time tax calculation only:

   ```
   - fm_bill taxation_switch 1
   ```

3. Save and close the file.
4. Stop and restart the CM.

Configuring Taxation in Pricing Center

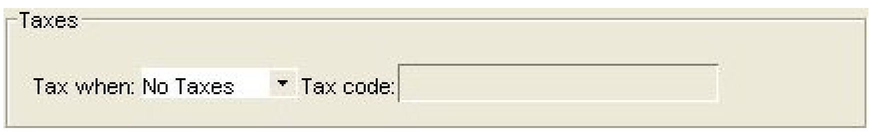
For each rate plan in Pricing Center, specify:

- The tax code to apply.
- When to apply the tax code: during real-time rating or billing.

To assign taxation to specific rate plans:

1. Open a real-time rate plan in Pricing Center.
2. In the Rate Plan Properties dialog box, specify when to calculate taxes and which tax code to apply as shown in Figure 6-1.

Figure 6-1 Rate Plan Properties Dialog Box



Configuring Taxation in loadpricelist

To assign the tax calculation method for a rate plan by using **loadpricelist**, create a price list XML file and set the rate plan's **tax_when** flag to one of the following:

- **NEVER**: Do not calculate or apply taxes.
- **NOW**: Calculate and apply taxes when the event occurs.
- **DEFER**: Defer tax calculation until the end of the billing cycle.
- **DYNAMIC**: Defer tax calculation until the end of the billing cycle but use the tax rate from the event time. For example, if an account purchases a product on June 15 but its billing cycle ends on June 30, BRM would perform the tax calculation on June 30 but use the tax rate from June 15.

For example, the following specifies to apply dynamic taxation to a rate plan:

```
<rate_plan tax_when="DYNAMIC">
...
</rate_plan>
```

After you create the price list XML file, load it into the BRM database by using the **loadpricelist** utility. For more information, see "Using the XML Pricing Interface to Create a Price List" in *BRM Setting Up Pipeline Pricing*.

Configuring Tax Calculation in ECE

You can also configure the default values for simple, fixed-rate taxes (for example, GST or VAT) directly in Oracle Communications Elastic Charging Engine.

For more information, see "Configuring Taxation in ECE" in *ECE Implementing Charging*.

7

Configuring Vertex Data Manager for Vertex Q Series

In Oracle Communications Billing and Revenue Management (BRM), you can configure Vertex Data Manager to calculate taxes.

Topics in this document:

- About Implementing Vertex Q Series Tax Calculation
- Installing the Vertex Software
- Configuring the Sales Tax Q Series Shared Libraries
- Configuring the Communications Tax Q Series Shared Libraries
- Configuring the Vertex DM for Sales Tax Q Series
- Configuring the Vertex DM for Communications Tax Q Series
- Configuring the CM for Vertex Tax Calculation
- Verifying That Taxes Are Being Calculated

About Implementing Vertex Q Series Tax Calculation

Vertex Data Manager (DM) connects BRM with the Vertex Tax Q Series database, which maintains current tax rates for various tax jurisdictions. Using Vertex, you can simplify how taxes are calculated and use the latest tax rates.

To calculate taxes using Vertex Tax Q Series, you install BRM Vertex Manager and then run the Vertex Data Manager.

> **Note**
>
> BRM audits tax calculation data, but does not support the Vertex auditing features.

Installing the Vertex Software

You install BRM Vertex Manager when you install BRM.

Install the Vertex Sales Tax Q Series and Communications Tax Q Series software on the same machine as the Vertex Data Manager (DM). The Vertex software must be installed, configured, and tested as indicated in the Vertex documentation before any attempt is made to configure the Vertex DM.

See "Additional BRM Software Requirements" in the *BRM Compatibility Matrix* for information on supported versions of the Vertex software.

See the Vertex documentation for installation instructions. You can install either package or both.

> **Note**
>
> The Sales Tax Q Series and Communications Tax Q Series database can reside in the same tablespace as the BRM database, but it is recommended that you create a new tablespace for it.

Write down the values you entered for the following when you installed the Vertex Tax Q Series software:

- Data Source For Register Database
- Server Name For Register Database
- User ID For Register Database
- Password For Register Database

These values are used later when you configure the **pin.conf** entries.

Configuring the Sales Tax Q Series Shared Libraries

Ensure the following libraries are in the *BRM_home***/lib** directory or in your **$LD_LIBRARY_PATH**. These are shared libraries that come with the Sales Tax Q Series software.

- **libvst.so**
- **libloc.so**
- **libqutil.so**

For more information about using Vertex toolkits, see the Vertex documentation.

Specifying Storage Manager Shared Library for Sales Tax Q Series

To configure the Vertex DM for Sales Tax Q Series, edit the *BRM_home***/sys/dm_vertex/pin.conf** entries in Table 7-1.

Table 7-1 pin.conf Entries for Vertex DM (Sales Tax Q Series)

Entry	Description
quantumdb_source	The schema where the STQ tables reside. **Note:** This parameter is required if the default tablespace for the user name and password parameters is not the same as the one that contains the STQ tables. For Indexed Sequential Access Method (ISAM) databases, this parameter specifies the ISAM data file directory.
quantumdb_server	The network identifier for the database on the server. For ISAM databases, this parameter should be commented out.
quantumdb_user	A valid Oracle user login name. For ISAM databases, this parameter should be commented out.
quantumdb_passwd	The user password.

Table 7-1 (Cont.) pin.conf Entries for Vertex DM (Sales Tax Q Series)

Entry	Description
quantum_sm_obj	The Storage Manager shared library that the DM uses to interact with a particular version of the Vertex STQ system. For example, use **dm_vertex_stq100** to interact with STQ v10.0.

To specify the appropriate Storage Manager shared library, set the value for **quantum_sm_obj** entry in the **pin.conf** file for Vertex DM to the particular version of Vertex STQ that you intend to use:

1. Open the Vertex DM configuration file (*BRM_home*/**sys/dm_vertex/pin.conf**).
2. Set the **quantum_sm_obj** entry, which is displayed in the following format in the file:

   ```
   - dm_vertex quantum_sm_obj ./dm_vertex_stqVersion${LIBRARYEXTENSION}
   ```

   replace *Version* with the STQ version, such as 100 for STQ 10.0.
3. Save and close the file.
4. Stop and restart the Vertex DM.

Configuring the Communications Tax Q Series Shared Libraries

Ensure the following libraries are in the *BRM_home*/**lib** directory or in your **$LD_LIBRARY_PATH**. These shared libraries come with the Vertex Communications Tax Q Series software.

- **libadm.so**
- **libcch.so**
- **libcfg.so**
- **libcli.so**
- **libctq.so**
- **libctz.so**
- **libdbcper.so**
- **libgeo.so**
- **libhsh.so**
- **libobj.so**
- **libreg.so**
- **librpt.so**
- **librte.so**
- **libutl.so**
- **libxmlparse.so**

Specifying Storage Manager Shared Library for Communications Tax Q Series

Table 7-2 lists the **pin.conf** entries in the *BRM_home***/sys/dm_vertex/pin.conf** file used to configure the Vertex DM for Communications Tax Q Series.

Table 7-2 pin.conf Entries for Vertex DM (Communications Tax Q Series)

Entry	Description
commtax_config_path	The location of the Communications Tax Q Series configuration file (**ctqcfg.xml**). The default location is *CTQ_home***/vertex/cfg**, where *CTQ_home* is the directory where you installed the Vertex software.
commtax_config_name	The Communications Tax Q Series configuration name. This name must match the configuration defined in the **ctqcfg.xml** file that is used with BRM. See the Vertex documentation for more information about defining configurations.
commtax_sm_obj	The Storage Manager shared library that the DM uses to interact with a particular version of the Vertex CTQ system. To use CTQ 3.02, specify **dm_vertex_ctq302**.

To specify the appropriate Storage Manager shared library, set the value for **commtax_sm_obj** entry in the **pin.conf** file for Vertex DM to the particular version of Vertex CTQ that you intend to use:

1. Open the Vertex DM configuration file (*BRM_home***/sys/dm_vertex/pin.conf**).
2. Set the **commtax_sm_obj** entry, which is displayed in the following format in the file:

   ```
   dm_vertex commtax_sm_obj ./dm_vertex_ctqVersion${LIBRARYEXTENSION}
   ```

   replace *Version* with the CTQ version, such as 302 for CTQ 3.02.
3. Save and close the file.
4. Stop and restart the Vertex DM.

Configuring the Vertex DM for Sales Tax Q Series

To configure the Vertex DM for Sales Tax Q Series, you specify the required entries in the *BRM_home***/sys/dm_vertex/pin.conf** file:

1. Add the QuantumDB password to the BRM client wallet using **pin_config_editor**:

   ```
   pin_config_editor -setconf -wallet wallet/client -parameter
   dm_vertex.quantumdb_passwd -pwd
   Enter the value:
   Enter Password for the wallet:
   Setting the entry dm_vertex.quantumdb_passwd is SUCCESSFUL
   ```

2. Open the Vertex DM configuration file (*BRM_home***/sys/dm_vertex/pin.conf**).
3. Change the following entries according to the instructions in the file:

   ```
   - dm_vertex quantumdb_source tablespace_for_Sales_Tax_C_Series_tables
   - dm_vertex quantumdb_server database_name
   ```

```
- dm_vertex quantumdb_user database_user
#- dm_vertex quantumdb_passwd database_user_password
```

> **Note**
>
> If you choose to keep your passwords in the **pin.conf** file rather than wallet, uncomment the following entry according to the instructions in the file:
>
> ```
> #- dm_vertex quantumdb_passwd database_user_password
> ```

where:

- The values for **quantumdb_source**, **quantumdb_server**, **quantumdb_user**, and **quantumdb_passwd** are the same values used when you installed the Sales Tax Q Series software. See "Installing the Vertex Software".
- **quantumdb_source** is required if the default tablespace for the user ID and password parameters is not the same tablespace that contains the Sales Tax Q Series tables.

> **Note**
>
> If the Vertex Sales Tax Q Series installation uses an ISAM database:
>
> – **quantumdb_source** should point to the ISAM data file directory (as opposed to the Oracle data source).
>
> – **quantumdb_server**, **quantumdb_user**, and **quantumdb_passwd** should be commented out.
>
> – **dm_n_fe** and **dm_n_be** should be set to **1** to prevent multithreading. The ISAM version of Vertex does not support multithreading.

4. Save and close the file.
5. Stop and restart the Vertex DM.

Configuring the Vertex DM for Communications Tax Q Series

To configure the Vertex DM for Communications Tax Q Series:

1. Open the Vertex DM configuration file (*BRM_home*/**sys/dm_vertex/pin.conf**).
2. Specify the path to the Communications Tax Q Series configuration file in the **commtax_config_path** entry.

   The following entry specifies the default path:

   ```
   - dm_vertex commtax_config_path CTQ_home/vertex/cfg
   ```
3. Specify the Communications Tax Q Series configuration name in the **commtax_config_name** entry.

   For example:

   ```
   - dm_vertex commtax_config_name CTQ_Test
   ```
4. Save and close the file.
5. Stop and restart the Vertex DM.

Configuring the CM for Vertex Tax Calculation

Configure CM for Vertex calculation by completing these tasks:

- Specifying Vertex DM Connection Entries in CM Configuration File
- Itemizing or Summarizing Taxes for Each Jurisdiction Level

Specifying Vertex DM Connection Entries in CM Configuration File

When you install BRM, you specify connection entries. You must change these entries if you change the Vertex database number, the hostname, or the port number of the Vertex DM.

To specify the Vertex DM connection entry in the CM configuration file:

1. Open the CM configuration file (*BRM_home*/**sys/cm/pin.conf**).
2. Edit the **dm_pointer** entry.

   ```
   - cm  dm_pointer  database  ip  hostname  port
   ```

   where:

   - *database* identifies the Vertex database number, such as 0.0.8.1.
   - *hostname* is the IP address or hostname of the computer on which the DM is installed.
   - *port* is the port number of the DM service.

   > **Note**
   >
   > The database number, hostname, and port number must match the values in the Vertex DM configuration file (*BRM_home*/**sys/dm_vertex/pin.conf**).

3. Edit the **vertex_db** entry:

   ```
   - fm_rate  vertex_db  database  /_tax_db  0
   ```

   where *database* is the number of the Vertex database. It must match the value entered in the **dm_vertex pin.conf** file.
4. Save and close the file.

You don't need to restart the CM to enable these entries.

Itemizing or Summarizing Taxes for Each Jurisdiction Level

You can show the details of the tax types returned by Communications Tax Q Series for each tax jurisdiction level. These details can be mapped to a general ledger (G/L) ID and be shown on the invoice.

Some of the Vertex CTQ codes are different from the internal BRM jurisdiction codes to which they are mapped. Table 7-3 lists the Vertex-to-BRM mappings whose codes differ.

Table 7-3 Numbering Differences in Vertex-to-BRM Jurisdiction Code Mapping

Jurisdiction	Vertex Code	BRM Code
Other Municipality	6	12
County District	7	10
City District	9	11

To itemize the taxes for each jurisdiction:

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

   ```
   pin_bus_params -r BusParamsRating bus_params_rating.xml
   ```

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

   ```
   <TaxReturnJuris>itemize</TaxReturnJuris>
   ```

   The default value is **summarize**.

   > **Caution**
   >
   > BRM uses the XML in this file to overwrite the existing instance of the **/config/business_params** object. If you delete or modify any other parameters in the file, these changes affect the associated aspects of the BRM configuration.

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

   ```
   pin_bus_params bus_params_rating.xml
   ```

   You should run this command from the *BRM_home*/**sys/data/config** directory, which includes support files used by the utility. To run it from a different directory, see "pin_bus_params" in *BRM Developer's Guide*.
7. Read the object with the **testnap** utility or the Object Browser to verify that all fields are correct.
   For general instructions on using **testnap**, see "Using the testnap Utility to Test BRM" in *BRM Developer's Guide*. For information on how to use Object Browser, see "Reading Objects" in *BRM Developer's Guide*.
8. Stop and restart the Connection Manager (CM).
   For more information, see "Starting and Stopping the BRM System" in *BRM System Administrator's Guide*.

Verifying That Taxes Are Being Calculated

After you have installed and configured the Vertex software, you can run the following test to verify that the Vertex software is calculating taxes.

1. Create a text file named **T502_40** with the following contents:

```
0 PIN_FLD_POID                        POID [0] 0.0.0.1 /account 1 1
0 PIN_FLD_END_T                     TSTAMP [0] (938224941) Fri Sep 24 19:02:21 1999
0 PIN_FLD_ACCOUNT_NO                   STR [0] "ROOT"
0 PIN_FLD_CURRENCY                     INT [0] 840
0 PIN_FLD_CURRENCY_NAME                STR [0] "USD"
0 PIN_FLD_TAXES                      ARRAY [0] allocated 20, used 10
1     PIN_FLD_TAX_CODE                 STR [0] "installVertex"
1     PIN_FLD_AMOUNT_TAXED         DECIMAL [0] 20
1     PIN_FLD_GL_ID                    INT [0] 0
1     PIN_FLD_SHIP_TO                  STR [0] "Cupertino;CA;95014;US;[408572,2,1]"
1     PIN_FLD_SHIP_FROM                STR [0] "Denver; CO; 80205;US;[303279,2,1]"
1     PIN_FLD_ORDER_ORIGIN             STR [0] ""
1     PIN_FLD_ORDER_ACCEPT             STR [0] ""
1     PIN_FLD_INTERNATIONAL_IND        INT [0] 1
1     PIN_FLD_LOCATION_MODE           ENUM [0] 2
1     PIN_FLD_ELAPSED_TIME          TSTAMP [0] (249)
0 PIN_FLD_VAT_CERT                     STR [0] ""
0 PIN_FLD_INCORPORATED_FLAG           ENUM [0] 0
0 PIN_FLD_RESIDENCE_FLAG              ENUM [0] 0
0 PIN_FLD_REGULATED_FLAG              ENUM [0]
```

2. Save the text file to the *BRM_home***/setup/scripts** directory.
3. Start **dm_vertex**.
4. Open the *BRM_home***/sys/cm.pinlog** file and verify that there are no errors.
5. Run **testnap** in the *BRM_home***/setup/scripts** directory to verify that the taxes are being calculated:

```
testnap
r T502_40 1
xop 502 0 1
```

> **Note**
>
> 502 is the opcode reference number for PIN_FLD_RATE_TAX_CALC.

8

Configuring Taxation Gateway for Vertex O Series

In Oracle Communications Billing and Revenue Management (BRM), you can configure Vertex O Series to calculate taxes using the Taxation Gateway.

Topics in this document:

About Implementing Vertex O Series Tax Calculation

Vertex O Series integrates BRM with Vertex's cloud tax service using the Taxation Gateway. To calculate taxes using Vertex Tax O Series, you install and run the BRM Taxation Gateway.

BRM maps tax data to Vertex REST API requests using the Mapper framework, sends the requests to the configured REST endpoints, and returns jurisdiction-level results for billing. You configure the Taxation Gateway properties in the **application.yaml** file. You do not install Vertex libraries with BRM; BRM uses REST to communicate with Vertex O Series.

> **Note**
>
> BRM audits tax calculation data but does not support the Vertex auditing features.

To implement Vertex O Series tax calculation:

1. Set up the Vertex O Series software. See "Setting Up the Vertex Software" for more information.
2. Configure your tax codes. See "Creating Tax Codes for Vertex O Series" for more information.
3. Configure your tax suppliers. See "Providing the Tax Supplier Data for Vertex O Series" for more information.
4. Configure the Taxation Gateway. See "Configuring the Taxation Gateway" for more information.
5. Configure your system to validate account addresses. See "Validating Addresses with Vertex O Series" for more information.

6. Specify the detail level to provide in invoices. See "Itemizing or Summarizing Taxes for Each Jurisdiction Level" for more information.
7. Test that taxes are calculated correctly. See "Verifying That Taxes Are Being Calculated" for more information.

You can also customize the way you calculate taxes, validate addresses, and the authenticator. See "Customizing the Taxation Gateway" for more information.

Setting Up the Vertex Software

Vertex O Series is a cloud service. You do not install Vertex software or databases on the BRM host. Instead, BRM connects to Vertex over REST using configured provider settings (base URI and OAuth credentials).

To set up Vertex O Series, ensure you have the following:

- A Vertex O Series tenant with API access enabled
- Network connectivity from BRM to the configured Vertex O Series base URI
- OAuth 2.0 credentials for Vertex O Series, stored and managed securely

See *BRM Compatibility Matrix* for information on supported versions of the Vertex software.

See the Vertex documentation for setting up your tenant and API access in Vertex O Series.

Configuring the Taxation Gateway

To configure the Taxation Gateway for Vertex O Series:

1. Edit the *BRM_home***/sys/taxation_gateway/Infranet.properties** file, and set the following values:
   a. Set the following parameter to the server port:
      ```
      infranet.server.portNr=16025
      ```
   b. Set the following parameter to the path to the Taxation Gateway configuration file:
      ```
      infranet.taxation.gateway.config.path=BRM_HOME/sys/taxation_gateway/
      application.yaml
      ```
      where *BRM_HOME* is the home directory for BRM.
2. Edit the Taxation Gateway configuration file (*BRM_home***/sys/taxation_gateway/application.yaml**) and set the following values:
   a. Change the following entries to define the Vertex O Series provider and endpoint:
      ```
      providers:
        - id: 6
          name: "Vertex"
          uri: baseURI
          authentication:
            manager:
      "com.oracle.communications.brm.taxation.authentication.GatewayTokenAuthentication
      Manager"
              configuration:
                path: "/oseries-auth/oauth/token"
                method: "POST"
                headers:
      ```

```
      Content-Type : "application/x-www-form-urlencoded"
      Accept : "application/json"
    payload:
      grant_type: "client_credentials"
      client_id: "${vertex.client.id}"
      client_secret: "${vertex.client.secret}"
    token_field_path: "$.access_token"
```

  b. Set the path to the mapper specifications in **mapping.specs.directory** to *BRM_home***/sys/taxation_gateway/mappings**.

3. Stop and restart the Taxation Gateway. Ensure the Taxation Gateway is running on the default port **16025**.

Validating Addresses with Vertex O Series

To validate addresses with Vertex O Series, you configure and use the Address Lookup endpoint through the Taxation Gateway.

To configure Vertex O Series address validation:

1. Open the Taxation Gateway configuration YAML file (*BRM_home***/sys/taxation_gateway/application.yaml**) and verify that the **address-lookup** endpoint is defined:
2. Verify the Address Lookup endpoint is defined:

```
providers:
  - id: 6
    name: "Vertex"
    uri: BaseURI
    endpoints:
      - id: "supplies"
...
        path: "/vertex-ws/v2/address-lookup"
        method: "POST"
        headers:
          Content-Type: "application/json"
          Accept: "application/json"
        mapper:
          request:
            - "request.validate-address"
          response:
            - "response.validate-address"
```

3. In your account creation or address-capture flow, enable address validation by calling the Vertex Address Lookup operation to validate the city, state, postal code, and country.
4. Stop and restart the Taxation Gateway if you changed the application configuration.

> **Note**
>
> BRM uses the PCM_OP_TAX_GW_VALIDATE_ADDRESS opcode to validate addresses. For more information, see "Validating Tax Jurisdictions Based On Address" in *BRM Opcode Guide*.

Itemizing or Summarizing Taxes for Each Jurisdiction Level

Vertex returns jurisdiction details (for example, STATE, COUNTY, CITY, DISTRICT) in the tax response, and BRM can display them itemized or summarized on invoices. To configure whether BRM itemizes or summarizes jurisdiction-level taxes returned by Vertex O Series, set the invoice presentation in the CM configuration file.

To configure the jurisdiction presentation:

1. Specify the entry in the *BRM_home*/**sys/cm/pin.conf file**:
   - To itemize taxes by jurisdiction level:

     ```
     fm_rate tax_return_juris itemize
     ```

   - To summarize taxes into a single amount:

     ```
     fm_rate tax_return_juris summarize
     ```

3. Save and close the file.

4. Stop and restart the CM.

Verifying That Taxes Are Being Calculated

After you have installed and configured the Vertex software, you can run the following test to verify that the Vertex software is calculating taxes.

1. Create a text file named **T502_O** with the following contents:

   ```
   0 PIN_FLD_POID POID [0] 0.0.0.1 /_tax_db -1 0
   0 PIN_FLD_FLAGS INT [0] 0
   0 PIN_FLD_END_T TSTAMP [0] (1763032566)
   0 PIN_FLD_BILL_OBJ POID [0] 0.0.0.1 /bill 1 0
   0 PIN_FLD_ACCOUNT_NO STR [0] "0.0.0.1-1"
   0 PIN_FLD_CURRENCY INT [0] 840
   0 PIN_FLD_CURRENCY_NAME STR [0] "USD"
   0 PIN_FLD_TAXPKG_TYPE ENUM [0] 6
   0 PIN_FLD_TAXES ARRAY [0] allocated 1, used 1
   1 PIN_FLD_ELEMENT_ID INT [0] 0
   1 PIN_FLD_TAX_CODE STR [0] "internet"
   1 PIN_FLD_AMOUNT_TAXED DECIMAL [0] 100
   1 PIN_FLD_LOCATION_MODE ENUM [0] 0
   1 PIN_FLD_SHIP_TO_ADDRESS SUBSTRUCT [0]
   2 PIN_FLD_COUNTRY STR [0] "US"
   2 PIN_FLD_ZIP STR [0] "94086"
   2 PIN_FLD_STATE STR [0] "CA"
   2 PIN_FLD_CITY STR [0] "SUNNYVALE"
   1 PIN_FLD_SHIP_FROM_ADDRESS SUBSTRUCT [0]
   2 PIN_FLD_COUNTRY STR [0] "US"
   2 PIN_FLD_ZIP STR [0] "94086"
   2 PIN_FLD_STATE STR [0] "CA"
   2 PIN_FLD_CITY STR [0] "SUNNYVALE"
   ```

   > **Note**
   >
   > Use the TAX_CODE that you mapped with TAXPKG_TAX_CODE in **config_taxcodes_map.xml**. For Vertex O Series, the Tax Code is **O**.

2. Save the text file to the *BRM_home***/setup/scripts** directory.
3. Start Taxation Gateway.
4. Open the *BRM_home***/sys/cm.pinlog** file and verify that there are no errors.

5. Run **testnap** in the *BRM_home***/setup/scripts** directory to verify that the taxes are being calculated:

```
testnap
r T502_0 1
xop 9441 - 0 1
```

> **Note**
>
> *9441* is the opcode reference number for PCM_OP_TAX_GW_CALCULATE_TAX.

Customizing the Taxation Gateway

You can customize the taxation gateway to change the way it calculates taxes and validates addresses. Additionally, you can use a custom authenticator instead of OAuth 2.0. To customize the taxation gateway, you extend the default Vertex O Series configurations using the **application.yaml** file and optional custom classes.

To customize the taxation gateway:

1. Decide what to customize: Authentication, Validating Addresses or Calculating Taxes.
2. Implement a custom policy class by using the **GatewayPolicyInterface**. Extend the policy class of the item that you want to customize:
   - For validating addresses, extend **VertexValidateAddressGatewayPolicy**, applied to the Tax Area lookup endpoint.
   - For calculating taxes, extend **VertexSuppliesCalculateTaxGatewayPolicy**, applied to the Supplies Tax Calculate endpoint.
   - For authentication, implement **GatewayAuthenticationManagerInterface** for custom authorization.
3. Override the following **GatewayPolicyInterface** methods to perform request validation before the call and response handling after the call:
   - Before Request Convert Hook using the method **beforeRequestConvert()**:

     ```
     FList beforeRequestConvert(
     GatewayContext context,
     FList requestFlist)
     throws GatewayException, EBufException
     {
     // ... custom business logic ...
     return requestFlist;
     }
     ```

   - Before Request Hook using the method **beforeRequest()**:

     ```
     Flist beforeRequest(
     GatewayContext context,
     FList requestFlist,
     JsonNode requestJson)
     throws GatewayException, EBufException
     {
     // ... custom business logic ...
     return requestJson;
     }
     ```

   - Before Response Convert Hook using the method **beforeResponseConvert()**:

```
Flist beforeResponseConvert(
GatewayContext context,
FList requestFlist,
JsonNode requestJson,
JsonNode responseJson)
throws GatewayException, EBufException
{
// ... custom business logic ...
return responseJson;
}
```

- Before Response Hook using the method **beforeResponse()**:

```
FList beforeResponse(
GatewayContext context,
FList requestFlist,
JsonNode requestJson,
JsonNode responseJson,
FList responseFlist)
throws GatewayException,
EBufException
{
// ... custom business logic ...
return responseFlist;
}
```

- When customizing authentication, override **GatewayAuthenticationManagerInterface** using the method **authenticate()**:

```
 authenticate(
int id,
HttpRequest.Builder builder)
throws GatewayException
{
// ... custom business logic ...
}
```

4. In the **application.yaml** file, set **providers[].endpoints[].hook** to your customized policy hook class for each customized endpoint.
5. Package your classes in a **.jar** file and add them to the runtime classpath using the TAXATION_GATEWAY_CLASSPATH_EXT environment variable.
6. Verify that the infranet.taxation.gateway.config.path in the **infranet.properties** file points to the **application.yaml** file, and ensure that the mapping.specs.directory path is set. Ensure the service port is set to **16025**.
7. Define your provider and endpoints in the **application.yaml** file by setting the provider ID, name, base URI, each endpoint's ID, path, method, headers, and the hook class name for custom policy hooks.
8. Configure authentication in the **application.yaml** file through the OAuth 2.0 **GatewayTokenAuthenticationManager** with its token endpoint, headers, payload, and **token_field_path**.
9. Create custom request and response mapping specifications for FLIST to JSON and JSON to FLIST, add the YAML files in the mapping.specs.directory, and reference them under each endpoint's mapper.request and mapper.response entries in the **application.yaml** file. For more information on creating custom mapper files, see "Configuring and Deploying Custom Mapper Files" in *BRM Cloud Native System Administrator's Guide*.
10. Configure routing in the **application.yaml** file with taxcode_groups so your tax codes map to the correct provider and endpoint at runtime, and add taxcode_mappings if you need static enrichments in the outgoing request payload.

11. Update the configuration property **providers[].endpoints[].hook** for the customized endpoints in the **application.yaml** file.
12. Review the HTTP client behavior and precision settings in the **application.yaml** file to meet your performance requirements.
13. Restart the Taxation Gateway. Verify that the configuration and custom **.jar** files load without errors by checking the taxation gateway logs in the PCM log path.
14. Validate the customization by calling the PCM_OP_TAX_GW_VALIDATE_ADDRESS and PCM_OP_TAX_GW_CALCULATE_TAX opcodes.

9

Tax Calculation Utilities

In Oracle Communications Billing and Revenue Management (BRM), you can learn about the different tax calculation utilities included.

Topics in this document:

- load_pin_ar_taxes
- load_tax_supplier

To learn about tax calculation, see "About Calculating Taxes".

load_pin_ar_taxes

Use this utility to load configurable tax information into the **/config/ar_taxes** object in the BRM database. See "Calculating Taxes for Accounts Receivable Actions".

Location

BRM_home/**bin**

Syntax

```
load_pin_ar_taxes [-v] [-h] [-f input_file.xml]
```

Parameters

-v
Displays information about successful or failed processing as the utility runs.

-h
Displays the utility's syntax and parameters.

-f *input_file.xml*
Specifies the name and location of the file that defines the configurable tax information. For example: **C:\taxconf\adjust_taxes.xml**. If you do not specify the **-f** parameter and a path, **load_pin_ar_taxes** looks for a file named **pin_config_ar_taxes.xml** in the directory from which you started the utility.

load_tax_supplier

Use the **load_tax_supplier** utility to load one or more tax suppliers into the BRM database. See "Creating Tax Suppliers".

Location

BRM_home/**bin**

Syntax

```
load_tax_supplier [-d | -v | -t | -h] pin_tax_supplier.xml
```

Parameters

-d
Creates a log file for debugging purposes. Use this parameter for debugging when the utility appears to have run with no errors, but the data has not been loaded into the database.

-v
Displays information about successful or failed processing as the utility runs.
To redirect the output to a log file, use the following command. Replace *filename*.**log** with the name of the log file:

```
load_tax_supplier pin_tax_supplier.xml -v > filename.log
```

> **Note**
>
> If a file with the same name exists, it is overwritten.

-t
Checks the validity of the XML file, but doesn't create an object.

-h
Displays the syntax and parameters for this utility.

-c
Creates a tax supplier object and configuration object pointed to it.

pin_tax_supplier
The name and location of the file that defines the tax suppliers. The default **pin_tax_supplier.xml** file is in *BRM_home*/**setup/scripts**.
If you do not run the utility from the directory in which the file is located, you must include the complete path to the file, for example:

```
load_pin_tax_supplier BRM_home/setup/scripts/pin_tax_supplier.xml
```

Results

If the **load_tax_supplier** utility doesn't notify you that it was successful, look in the **default.pinlog** file to find any errors. This file is either in the directory from which you started the utility or in a directory specified in the utility configuration file.