

Oracle® Communications Billing and Revenue Management

Calculating Taxes



Release 15.0

F86217-01

February 2024

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Communications Billing and Revenue Management Calculating Taxes, Release 15.0

F86217-01

Copyright © 2017, 2024, Oracle and/or its affiliates.

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

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

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

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

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

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

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

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

Contents

Preface

Audience	vi
Documentation Accessibility	vi
Diversity and Inclusion	vi

1 About Calculating Taxes

About Calculating Taxes	1-1
Impacts of Tax Calculation	1-2
About Tax Exemptions	1-2

2 Calculating Taxes for Accounts Receivable Actions

About Taxation for Adjustments, Disputes, and Settlements	2-1
About Adjustment Tax and Billing Cycles	2-1
About Dispute and Settlement Taxes and Billing Cycles	2-2
Configuring Tax Calculation for Adjustments, Disputes, and Settlements	2-2
Configuring the Default Tax Treatment for Customer Center	2-4
Configuring the Default Tax Method for Account Adjustments	2-4
Configuring Itemized Tax Information	2-5

3 Creating Tax Codes

About Creating Tax Codes	3-1
Creating Tax Codes Using ImportExportPricing	3-1
Creating Tax Codes for a Flat Tax or Custom Implementation	3-2
Creating Tax Codes for Vertex	3-4
Creating Tax Selectors Using ImportExportPricing	3-6
Creating an XML File for Tax Selectors	3-6

4 Creating Tax Suppliers

About Tax Suppliers	4-1
---------------------	-----

Providing the Tax Supplier Data with Each Transaction	4-2
Allowing Your Sales and Use Tax Software to Determine Tax Supplier Data	4-4
Specifying Divisions for Tax Suppliers	4-6
Defining a Default Ship-From Locale	4-7
Overriding Shipped-To with the Real-Time Location	4-7

5 Creating Tax Exemptions

About Creating Tax Exemptions for an Account	5-1
About Creating Tax Exemptions for Products	5-1
Creating Product-Level Tax Exemption Codes Using ImportExportPricing	5-2
Creating an XML File for Tax Exemption Codes	5-2
Creating Tax Exemption Selectors Using ImportExportPricing	5-4
Creating an XML File for Tax Exemption Selectors	5-5

6 Configuring Tax Calculation Options

Configuring Tax Calculation for Account Groups	6-1
Validating Customer Addresses	6-2
Reporting Zero Tax Amounts	6-3
Reporting Taxes of Zero Percent	6-3
Enabling and Disabling Taxation Globally	6-4
Configuring Taxation in Pricing Center	6-4
Configuring Taxation in loadpricelist	6-5
Configuring Tax Calculation in ECE	6-5

7 Configuring Vertex Manager

About Implementing Vertex Tax Calculation	7-1
Installing the Vertex Software	7-1
Configuring the Sales Tax Q Series Shared Libraries	7-2
Specifying Storage Manager Shared Library for Sales Tax Q Series	7-2
Configuring the Communications Tax Q Series Shared Libraries	7-3
Specifying Storage Manager Shared Library for Communications Tax Q Series	7-4
Configuring the Vertex DM for Sales Tax Q Series	7-4
Configuring the Vertex DM for Communications Tax Q Series	7-5
Configuring the CM for Vertex Tax Calculation	7-6
Specifying Vertex DM Connection Entries in CM Configuration File	7-6
Specifying whether to Validate ZIP Codes	7-7
Itemizing or Summarizing Taxes for Each Jurisdiction Level	7-7

8 Tax Calculation Utilities

load_pin_ar_taxes

8-1

load_tax_supplier

8-1

Preface

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.

Documentation Accessibility

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

Access to Oracle Support

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

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

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](#)
- [Impacts of Tax Calculation](#)
- [About Tax Exemptions](#)

About Calculating Taxes

BRM uses tax codes to calculate the taxes for charges and discounts. Tax codes define the tax percentage to apply to a charge, the jurisdiction (state or local), and how to calculate the tax amount. When you create product offerings, you can specify the tax code to apply to charges, discounts, and chargines. For example, you can use one tax code for telephone handsets and another for online service subscriptions.

When configuring charges, you can also specify when to calculate the taxes. You can calculate taxes at these times:

- **During rating:** In this configuration, taxes are calculated when the event is rated and added to the customer's account balance. This way, you always have an accurate reading of a customer's account balance at any time in the accounting cycle.
- **During billing:** In this configuration, taxes are calculated at billing time. Deferring tax calculation to the billing process reduces rounding errors because all events of the same type are taxed together. For example, BRM calculates taxes for the total usage fees rather than individual usage events.
- **During billing, but using the event time:** In this configuration, taxes are calculated at billing time using the tax rate at the time the event occurred. This method provides the benefits of billing-time taxation while allowing you to change tax rates during a billing cycle. For example, assume a recurring charge's tax rate changes from 2% to 3% on May 15. If a billing cycle ends on May 30, BRM will use the 2% rate to calculate the charge's April 30 through May 14 taxes and the 3% rate to calculate the charge's May 15 through May 29 taxes.

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 for multiple jurisdictions. With this method, you run the Vertex Data Manager to communicate with the Vertex tax database.

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 effectively reverse whatever tax was levied on the original item. See "Accounts Receivable" in *Billing Care Online Help*.
- You can record taxes as general ledger data. To do so, 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 amount in the rerated events during the subsequent bill run. The rerated tax appears on the invoice for the next bill run. You can configure BRM to apply deferred taxes during rerating. This enables BRM to include the deferred tax amounts on corrected invoices.
- You can define tax exemptions for customers in Billing Care or Customer Center. For example, you can specify if an account is exempt from city taxes. You can also specify the exemption percentage, such as 10% of the amount is not taxed or 100% for a total exemption.

Vertex Communications Tax Q Series does not support partial tax exemptions in percentage. It 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 amount of tax your customers owe.

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 tax exemption codes to products based on account, service, event, or profile attributes.

If you define a tax exemption at both the account and product level for the same jurisdiction (for example, federal), BRM uses the account-level exemption.

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

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](#)
- [About Adjustment Tax and Billing Cycles](#)
- [About Dispute and Settlement Taxes and Billing Cycles](#)
- [Configuring Tax Calculation for Adjustments, Disputes, and Settlements](#)
- [Configuring Itemized Tax Information](#)

About Taxation for Adjustments, Disputes, and Settlements

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

You can specify whether to perform an adjustment, dispute, or settlement with or without taxes at the account, bill, item, and event level. For taxable account, bill, and item adjustments, BRM uses information supplied by a tax configuration file to calculate the tax reversal for the account.

If you include taxes in an A/R action and:

- Billing *has* already occurred: Taxes are always calculated when you create the adjustment, dispute, or settlement.
- Billing *has not* occurred: The tax amount is applied only if the item has real-time taxes. If the item has deferred taxes, the tax amount is calculated and applied at billing time.

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

- Configure a separate item for taxes exclusively. See "Cumulative Custom Item for Taxes" in *BRM Configuring and Running Billing*.
- All events associated with an item must use the same tax method: tax now or tax deferred. Nontaxable events are valid in either taxation scenario.

About Adjustment Tax and Billing Cycles

If the adjustment occurs at the account, subscription service, or member service level, BRM uses the total amount of the adjustment without consideration of the specific events that are included in the adjustment.

For bill, item, and event 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 that it will transfer for taxation during billing. BRM also proportions the deferred amount for partial adjustments according to how much of an adjustment was actually granted.

About Dispute and Settlement Taxes and Billing Cycles

For bill and item disputes and settlements, real-time taxes are applied when you create a dispute or settlement. When processing a dispute or settlement with deferred taxes, BRM determines whether the original item has been billed. If billing has not yet occurred, BRM has not levied the tax for the original item. In this case, there is no need to apply the taxes separately because the dispute or settlement will have acted on the balance of the original item before BRM ever calculates the taxes for that item.

However, if the dispute or settlement is opened against an item already billed, the customer has already been charged tax. In this case, BRM must specifically accommodate the dispute or settlement tax. BRM defers applying the tax reversal amount until you run billing.

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

Configuring Tax Calculation for Adjustments, Disputes, and Settlements

Depending on tax law, jurisdictions, and other factors, you may need to specify whether to apply a tax reversal when performing certain 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 for this configuration, you must load a **/config/ar_taxes** object that 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, you can configure a default tax treatment for adjustments, disputes, and settlements. Customer Center uses this configuration to determine whether these activities always have a tax reversal, never have a tax reversal, or have a tax reversal at the CSR's discretion. See "[Configuring the Default Tax Treatment for Customer Center](#)".
- If you don't use Customer Center to make adjustments, you can edit the CM **pin.conf** file to configure BRM to specify that tax reversals occur at the time account adjustments are created rather than deferring them until billing. See "[Configuring the Default Tax Method for Account Adjustments](#)".

If you use Customer Center, the tax reversal entry in the **pin.conf** file is ignored. Instead, you use the Customer Center Configurator to set up the tax treatment.

Before you load the tax configuration, you must first load the tax suppliers. See "[About Tax Suppliers](#)".

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

- The tax suppliers you specify in the **pin_config_ar_taxes.xml** file must be listed in the **tax_supplier_map** file.
- The tax code you use in the **pin_config_ar_taxes.xml** file must be present in the **taxcodes_map** file. The tax code is mapped to a package that calculates tax for that charge offer type, and can be a numeric or 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 all standard XML formatting rules. The following example shows a typical **pin_config_ar_taxes.xml** file.



Note:

The **<Name>** field in the sample 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 `robg` 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 are using Customer Center as your CRM application, you can configure a default tax treatment for adjustments, disputes, and settlements. Customer Center uses this configuration to determine whether these activities always have a tax reversal, never have a tax reversal, or have a tax reversal at the CSR's discretion. Configuring a default tax treatment can help you enforce uniform tax treatments for a given location and eliminate the need for a decision by the CSR.

You configure the tax treatment on the **Balance** tab in the Customer Center Configurator. This configuration overrides any tax treatment specification in the `pin.conf` file. To set up tax treatment for adjustments, disputes, and settlements, select one of the following options 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 an account adjustment is created.
- **Dynamic taxation:** BRM defers tax calculation for all account adjustments until the end of the billing cycle. However, it calculates the tax using the tax rate at the time of the account adjustment. For example, if an account creates an adjustment on June 15 and with a billing date of June 30, BRM calculates the tax on June 30 using the tax rate from June 15. 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:**

To ensure tax calculations are performed correctly, all events associated with an item must use the same tax method.

3. Save and close the file.

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

Configuring Itemized Tax Information

If you choose, you can save tax information at the item level. To do this, set the **ItemizedTaxCalculation** business parameter to **enabled**. To configure this feature, run the **pin_bus_params** utility to change the **ItemizedTaxCalculation** business parameter. See "pin_bus_params" in *BRM Developer's Guide* for more information about this utility.

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
```

3. In the file, 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.
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.

When you have enabled itemized tax calculation, the following information can be retrieved for each *item* on a bill using the **PCM_OP_AR_GET_ITEM_DETAIL** opcode:

- Settled tax amount
- Settled taxed amount
- Settled nontaxed amount

To enable itemized taxes during account adjustment, set the value of **tax_now** in *BRM_home/sys/cm/pin.conf* to **3**. See "[Configuring the Default Tax Method for Account Adjustments](#)" for more information about setting this parameter. When this parameter is set to **3**:

- When the adjustment is made at the account level with tax, the behavior remains the same as when the feature is not enabled: that is, the tax will not be calculated, **/event/billing/adjustment/account** will only contain the charge value impact, and the adjustment amount will remain unallocated.
- When the adjustment is allocated to an item, the tax will be calculated proportionately based on the tax applied for the destination item. A new **/event/billing/adjustment/account** will be created to record the tax billing impact, and the old adjustment event will be linked in the PIN_FLD_LINK_OBJ field of the new adjustment event.

If you have itemized taxation enabled and an event is rated with event-time tax, the event will contain one balance impact for the charge amount and another for the tax. The PIN_FLD_AMOUNT_TAXED and PIN_FLD_TAX fields in the **/item** will be updated accordingly.

If you have itemized taxation enabled and a bill or item-level adjustment occurs, the following processing will occur:

- If the adjustment input list contains the PIN_AR_WITH_TAX flag, then the tax will be calculated on the adjustment amount.
- The proportionate tax amount for the items will be calculated from the PIN_FLD_TAXES field in PIN_FLD_TAX_DATA, instead of using the tax percent and tax code from **/config/ar/taxes**. The adjusted tax amount will be updated in the PIN_FLD_AMOUNT_TAX_ADJUSTED field of PIN_FLD_TAXES and then updated in PIN_FLD_TAX_DATA.

Items created before this feature is enabled will not contain the itemized information. After the feature has been enabled, the itemized information may be inaccurate for the billing cycle during which the feature is activated, because of events that occurred before the feature was enabled.

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](#)
- [Creating Tax Codes Using ImportExportPricing](#)
- [Creating Tax Selectors Using ImportExportPricing](#)

About Creating Tax Codes

Tax codes define which tax to apply to a charge. For example, a telephone handset uses a different tax code than an online service subscription. The tax code identifies 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](#)".

After you finish creating tax codes, you can associate them with your charges, discounts, and chargseshares.

You can also assign tax codes to charges, discounts, and chargseshares using tax selectors. Tax selectors decide 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 directory in which the PDC software is installed. The sample file contains examples of tax codes for calculating taxes based on a custom implementation or Vertex. For more information about XML elements specific to Vertex or custom tax codes, see:
 - [Creating Tax Codes for Vertex](#)

- [Creating Tax Codes for a Flat Tax or Custom Implementation](#)
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 also 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 also publishes the tax codes to the real-time rating engine (RRE) and ECE.

 **Note:**

You cannot publish tax codes to the batch rating engine (BRE).

3. Map your tax codes to G/L IDs in ECE. Do the following for each tax code you configured in *MyTaxCodes.xml*:
 - a. Access the ECE configuration MBeans in a JMX editor, such as JConsole. See "Accessing ECE Configuration MBeans" in *ECE Implementing Charging* for more information.
 - b. Expand the **ECE Configuration** node.
 - c. Expand **charging.taxation**.
 - d. Expand **Operations**.
 - e. Click **addTaxDetails**.
 - f. Specify values for the following attributes:
 - **taxCode**: Enter a tax code. The tax code must match a value in the **<code>** element of your *MyTaxCodes.xml* file.
 - **taxGllid**: Enter the General Ledger ID to use for the tax impact.

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>
  <taxCodeValidityPeriods>
    <validFrom>20220101</validFrom>
    <validTo>20230120</validTo>
  <taxCodeMaps>
    <description>TRS1</description>
  </taxCodeMaps>
</taxCodeValidityPeriods>
</taxPackageType>
</code>
</obsolete>
</priceListName>
</name>
</taxCodes>
```



```

        <percent>4.0</percent>
        <taxJurisdictionLevel>FEDERAL</taxJurisdictionLevel>
        <taxJurisdictions>US</taxJurisdictions>
        <taxRuleType>TAX</taxRuleType>
    </taxCodeMaps>
</taxCodeValidityPeriods>
<taxCodeValidityPeriods>
    <validFrom>20240101</validFrom>
    <validTo>20300120</validTo>
    <taxCodeMaps>
        <description>TRS2</description>
        <percent>8.0</percent>
        <taxJurisdictionLevel>LOCATION</taxJurisdictionLevel>
        <taxJurisdictions>EU</taxJurisdictions>
        <taxRuleType>INCLUSIVE</taxRuleType>
    </taxCodeMaps>
    <taxCodeMaps>
        <description>TRS3</description>
        <percent>8.0</percent>
        <taxJurisdictionLevel>LOCATION</taxJurisdictionLevel>
        <taxJurisdictions>ASIA</taxJurisdictions>
        <taxRuleType>NCS</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 <i>yyyymmdd</i> 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 <i>yyyymmdd</i> format.
description	A brief description of 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 .

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

Element	Description
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.
taxRuleType	<p>Determines how taxes will be computed. Values are:</p> <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> 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. <div style="border: 1px solid #0070C0; padding: 10px; margin: 10px 0;"> <p> Note: Inclusive taxes are not supported by Oracle Communications Elastic Charging Engine (ECE).</p> </div> <ul style="list-style-type: none"> 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: <ul style="list-style-type: none"> tax1 = 10% @ 100.00 = 10.00 tax2 = 20% @ 100.00 = 20.00

Creating Tax Codes for Vertex

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

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

```
<taxCodes>
  <name>VQTaxCode</name>
  <priceListName>Default</priceListName>
  <obsolete>>false</obsolete>
  <code>VQTaxCode</code>
  <taxPackageType>VERTEX_QUANTUM</taxPackageType>
  <taxCodeValidityPeriods>
    <taxCodeMaps>
```

```

        <transType_CategoryCode_Rate>01</transType_CategoryCode_Rate>
        <transSubType_ServiceCode>01</transSubType_ServiceCode>
        <salesIndicator>SALE</salesIndicator>
    </taxCodeMaps>
</taxCodeValidityPeriods>
</taxCodes>
<taxCodes>
    <name>VCTaxCode</name>
    <priceListName>Default</priceListName>
    <obsolete>>false</obsolete>
    <code>VCTaxCode</code>
    <taxPackageType>VERTEX_COMMTAX_21</taxPackageType>
    <taxCodeValidityPeriods>
        <taxCodeMaps>
            <transType_CategoryCode_Rate>4</transType_CategoryCode_Rate>
            <transSubType_ServiceCode>2</transSubType_ServiceCode>
            <salesIndicator>SALE</salesIndicator>
        </taxCodeMaps>
    </taxCodeValidityPeriods>
</taxCodes>
    
```

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

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

Element	Explanation
code	The tax code. A unique alphanumeric value that defines categories with different tax treatments.
taxPackageType	The tax package you use: <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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.

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"?><pd:PricingObjectsJXB
xmlns:pd="http://xmlns.oracle.com/communications/platform/model/pricing">
<taxSelector xmlns:pd="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>
  </validityPeriod>
  <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 3-3 describes the important XML elements specific to tax selectors.

Table 3-3 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](#)
- [Providing the Tax Supplier Data with Each Transaction](#)
- [Allowing Your Sales and Use Tax Software to Determine Tax Supplier Data](#)
- [Defining a Default Ship-From Locale](#)
- [Overriding Shipped-To with the Real-Time Location](#)

About 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. Vertex uses the tax supplier to calculate taxes.

Tax supplier data can include the following information:

- 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 numbers are derived from call details records (CDRs).

**Note:**

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

You specify how to supply tax supplier data to your tax 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.

You can also configure a default ship-from address in case BRM cannot find any tax supplier. See "[Defining a Default Ship-From Locale](#)".

Providing the Tax Supplier Data with Each Transaction

If your business uses only one tax supplier, you can define a default tax supplier to use 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 tax supplier defined in the charge offer 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 of defining tax suppliers, you 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 the `pin_tax_supplier.xml` file. To modify or remove existing tax suppliers, edit or remove their information in the file before loading it.

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

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 <code>tax_supplier_map</code> 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: <i>city;state;zip_code;country</i>
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 Vertex Manager " 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.
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 working in the same directory as the **pin_tax_supplier.xml** file, include the complete path to the file.

2. Stop and restart the CM.

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

If your business has customers in multiple countries or operates under other conditions where complex tax laws are involved, you should allow your sales and use tax software to choose the correct tax supplier. The disadvantage is that you need to include every BRM charge offer in the **tax_supplier_map** file and update the file 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 <i>PDC Creating Product Offerings</i> .
ShipTo	The address the product is shipped to (your customer's location). Enter the address in this format: <i>city;state;ZIP;country</i> You can leave portions of this string blank, but you must include the semicolons for each address element. Examples: <ul style="list-style-type: none"> • ;;;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.
Company ID	The company ID specified in the Vertex TDM (Tax Decision Maker) or company file. This name is also entered 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.

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

Entry	Description
ShipFrom	<p>The address the product is shipped from. Enter the address in this format:</p> <p><i>city;state;ZIP;country</i></p> <p>You can leave portions of this string blank, but you must include the semicolons for each address element.</p> <p>Examples:</p> <ul style="list-style-type: none"> • ;;US • ;CA;95003;US <p>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.</p> <p>See the Vertex documentation for correct state and country entries.</p>
Regulated	<p>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.</p>

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

Some companies operate divisions with locations that can impact the tax calculation. Use the **Business Loc** column in the **tax_supplier_map** file to add information about divisions for tax suppliers. The data in this column is mapped to the **Division** field in Sales Tax Q Series. This data should match what is 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.

Defining a Default Ship-From Locale

This option provides a ship-from locale in case a tax supplier cannot be found. You need to 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, and all subsequent tax calculations use the new address. You do not need to restart the CM.

Overriding Shipped-To with the Real-Time Location

This option overrides the shipped-to locale with your customers' 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 the following opcodes with the PIN_FLD_LOCALE input field set to "*City;State;Zipcode;Country*":

- PCM_OP_SUBSCRIPTION_PURCHASE_DEAL
- PCM_OP_SUBSCRIPTION_PURCHASE_PRODUCT
- PCM_OP_PYMT_TOPUP

Alternatively, you can 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.
validFrom	The start date of the validity period for the tax exemption code in <i>yyyymmdd</i> 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 <i>yyyymmdd</i> 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"?>
<pd: PricingObjectsJXB xmlns:pd="http://xmlns.oracle.com/communications/platform/
model/pricing">
<taxExemptionSelector xmlns:pd="http://xmlns.oracle.com/communications/platform/
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>
        <separator>;</separator>
        <fieldName>TelcoGprs.GPRS_INFO.ROUTING_AREA</fieldName>
        <fieldValue>National</fieldValue>
        <fieldKind>PRODUCT_SPEC_FIELD</fieldKind>
      </fieldToValueExpression>
      <fieldToValueExpression>
        <operation>EQUAL_TO</operation>
        <separator>;</separator>
        <fieldName>EventSessionTelcoGprs.TELCO_INFO.PRIMARY_MSID</fieldName>
        <fieldValue>PMSID10248112193</fieldValue>
        <fieldKind>EVENT_SPEC_FIELD</fieldKind>
      </fieldToValueExpression>
      <fieldToValueExpression>
        <operation>IN_LIST</operation>
        <separator>;</separator>
        <fieldName>EventBillingProductFeeCycleCycle_forward_monthly.CYCLE_INFO.CYCLE_START_T</
        fieldName>
        <fieldValue>20100101</fieldValue>
        <fieldKind>EVENT_SPEC_FIELD</fieldKind>
      </fieldToValueExpression>
      <fieldToValueExpression>
        <operation>IN_LIST</operation>
        <separator>;</separator>
        <fieldName>EventBillingProductFeeCycleCycle_forward_monthly.CYCLE_INFO.CYCLE_END_T</
        fieldName>
        <fieldValue>20301231</fieldValue>
      </fieldToValueExpression>
    </rule>
  </validityPeriod>
</taxExemptionSelector>
</pd: PricingObjectsJXB>
```

```

        <fieldKind>EVENT_SPEC_FIELD</fieldKind>
    </fieldToValueExpression>
    <fieldToValueExpression>
        <operation>EQUAL_TO</operation>
        <separator>;</separator>
        <fieldName>AnyEvent.USAGE_TYPE</fieldName>
        <fieldValue>Downlink</fieldValue>
        <fieldKind>EVENT_SPEC_FIELD</fieldKind>
    </fieldToValueExpression>
    <fieldToValueExpression>
        <operation>EQUAL_TO</operation>
        <separator>;</separator>
        <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>
        <separator>;</separator>
        <fieldName>TelcoGprs.GPRS_INFO.ROUTING_AREA</fieldName>
        <fieldValue>SouthWest</fieldValue>
        <fieldKind>PRODUCT_SPEC_FIELD</fieldKind>
    </fieldToValueExpression>
    <fieldToValueExpression>
        <operation>EQUAL_TO</operation>
        <separator>;</separator>
        <fieldName>EventSessionTelcoGprs.TELCO_INFO.PRIMARY_MSID</fieldName>
        <fieldValue>PMSID10265018152</fieldValue>
        <fieldKind>EVENT_SPEC_FIELD</fieldKind>
    </fieldToValueExpression>
    <fieldToValueExpression>
        <operation>IN_LIST</operation>
        <separator>;</separator>

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

    <fieldName>EventBillingProductFeeCycleCycle_forward_monthly.CYCLE_INFO.CYCLE_END_T</fieldName>
        <fieldValue>20321231</fieldValue>
        <fieldKind>EVENT_SPEC_FIELD</fieldKind>
    </fieldToValueExpression>
    <fieldToValueExpression>
        <operation>EQUAL_TO</operation>
        <separator>;</separator>
        <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). However, there are two ways to calculate taxes. Taxes can be calculated for each individual nonpaying child bill unit and can be listed as separate items on the parent bill, or taxes can be consolidated into a single item for both the parent and nonpaying bill units.

You specify how BRM calculates taxes for account groups by entering one of the following values in the **cycle_tax_interval** entry in the Connection Manager (CM) configuration (**pin.conf**) file:

- **accounting:** (Default) BRM calculates taxes separately for the parent and each nonpaying bill unit and then lists the taxes as separate items on the parent bill.
- **billing:** BRM rolls activities for each nonpaying bill unit into the parent account and calculates taxes for the parent account only. The single tax item on the parent account includes taxes from both the parent and nonpaying bill units. This option improves overall performance of BRM tax computation.

Note:

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

To specify how to calculate taxes for account groups:

1. Open the CM configuration file (*BRM_home/sys/cm/pin.conf*).
2. Edit the **cycle_tax_interval** entry:
 - To forward the tax from the child account to the parent account, enter **billing**. BRM calculates taxes for the parent account only, but the single tax item on the parent account includes taxes from both the parent and child accounts.
 - To calculate the taxes separately for the parent and child accounts, enter **accounting**. BRM lists the taxes as separate items on the parent bill.

For example:

```
fm_bill cycle_tax_interval billing
```

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

Validating Customer Addresses

When taxes are calculated for customers in the United States, the tax rate is partially determined by the customer's ZIP code. If the customer's ZIP code is not correct and fails validation, or it is for 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 specifying which package to use for the **tax_valid** entry in the CM **pin.conf** file. 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.

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.

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 a zero amount is returned from taxation, BRM does *not* report the tax amount and the TAX_JURISDICTIONS array is not created for the event. However, legal requirements in some jurisdictions demand that zero tax amounts be reported.

To configure BRM to report zero taxes:

1. Open the CM configuration file (*BRM_home/sys/cm/pin.conf*).
2. Uncomment the **fm_rate include_zero_tax** entry and change the value of the entry to **1**:

```
- fm_rate include_zero_tax 1
```

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

If multiple taxes, including zero taxes, are reported from the same jurisdiction, they are aggregated by default. This may hide the effect of reporting zero taxes. To prevent zero taxes from being hidden in this way, you should configure BRM to itemize taxes by jurisdiction.

To configure BRM to itemize all taxes by jurisdiction:

1. Open the CM configuration file (*BRM_home/sys/cm/pin.conf*).
2. Change the value of the following entry to **itemize**:

```
- fm_rate tax_return_juris itemize
```

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

Reporting Taxes of Zero Percent

When a charge's tax rate is zero percent, BRM reports a **0** tax percentage rate and sets the PIN_FLD_BAL_IMPACTS to **0** by default. To configure this feature, run the **pin_bus_params** utility to change the **exclude_percent_for_zero_tax** 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 Manager 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 **exclude_percent_for_zero_tax** parameter:

```
<exclude_percent_for_zero_tax>value</exclude_percent_for_zero_tax>
```

where *value* is:

- **enabled** to have BRM report a zero tax percentage amount.
 - **disabled** to have BRM 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 turn BRM taxation on and off globally by using the **taxation_switch** entry in the CM configuration file (*BRM_home/sys/cm/pin.conf*). 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:

Because deferred taxation procedures are triggered even if BRM is not configured to use taxation features, turning taxation off can improve performance. If BRM is configured for taxation, you can turn taxation off and on for testing.

To enable or disable taxation globally:

1. Open the Connection Manager (CM) configuration file (*BRM_home/sys/cm/pin.conf*).
2. Set the value of the **taxation_switch** entry:
 - **0**: Tax calculation is entirely disabled.
 - **1**: Only real-time tax calculation is enabled.
 - **2**: Only deferred (cycle-time) tax calculation is enabled.
 - **3**: (Default) Both real-time and deferred tax calculations are enabled.

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

You can specify the following for each rate plan that you create in Pricing Center:

- 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

The screenshot shows a dialog box titled 'Taxes'. Inside, there is a label 'Tax when:' followed by a dropdown menu currently showing 'No Taxes'. To the right of the dropdown is a label 'Tax code:' followed by an empty text input field.

For more information on how to specify taxation in a rate plan, see Pricing Center Help.

Configuring Taxation in loadpricelist

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

- **NEVER**: Does not calculate and apply taxes.
- **NOW**: Calculates and applies taxes when an event occurs.
- **DEFER**: Defers tax calculation for an account until the end of its billing cycle.
- **DYNAMIC**: Defers tax calculation until the end of a billing cycle, but uses the tax rate from the time the event occurred. 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, you load it into the BRM database using the **loadpricelist** utility. For more information, see "Using the XML Pricing Interface to Create a Price List" in *BRM Configuring Pipeline Rating and Discounting*.

Configuring Tax Calculation in ECE

ECE supports a fixed tax rate (a flat-rate taxation also known as GST or VAT). In ECE, you can apply a tax on both charges and alterations (discounts).

To configure taxation in ECE:

1. Access the ECE configuration MBeans in a JMX editor, such as JConsole. See "Accessing ECE Configuration MBeans" in *ECE Implementing Charging* for more information.
2. Expand the **ECE Configuration** node.
3. Expand **charging.taxation**.
4. Expand **Operations**.
5. Click **addTaxDetails**.
6. Specify values for the following attributes:

 **Note:**

These attributes are mandatory. You must set all of them when configuring taxation.

- **taxCode:** Enter the tax code used by the charge offer or discount offer to which the tax applies.
The tax code is used by charge offers and discount offers to point to the tax rate that must be applied when a usage request is processed for the charge offer or discount offer.
Enter the same tax code entered in PDC or Pricing Center when the taxation section of the charge offer and discount offer was defined.
 - **taxRate:** Enter the tax rate to apply.
For example, entering **0.20** applies a 20% tax on the total usage impact.
 - **taxGld:** Enter the General Ledger ID used for the tax impact.
7. Specify an additional **taxCode**, **taxRate**, and **taxGld** value for each charge offer or discount offer to which a tax applies.

7

Configuring Vertex Manager

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

Topics in this document:

- [About Implementing Vertex 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 Tax Calculation

Vertex Manager connects BRM with the Vertex tax calculation 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, 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 Sales 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.

Use files with the extension **.so** for Linux:

- **libvst**
- **libloc**
- **libqutil**

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/sysl* **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.

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

Entry	Description
quantumdb_passwd	The user password.
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. Locate the **quantum_sm_obj** entry displayed in the following format in the file:


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

 where *Version* represents 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 <i>CTQ_home/vertex/cfg</i> , where <i>CTQ_home</i> 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. Locate the **commtax_sm_obj** entry displayed in the following format in the file:

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

where *Version* represents 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 **pin.conf** 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 **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](#)
- [Specifying whether to Validate ZIP Codes](#)
- [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 Data Manager, 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 database number specified in the Vertex DM configuration file.

4. Save and close the file.

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

## Specifying whether to Validate ZIP Codes

If there is an error in an account's ZIP code, Sales Tax Q Series returns a tax amount of 0. If you set up Sales Tax Q Series to validate ZIP codes at account creation, you ensure that taxes are calculated correctly. This option specifies that Sales Tax Q Series checks the city, state, and ZIP code when an account is created. If the ZIP code is invalid, account creation cannot be completed.

 **Note:**

- Communications Tax Q Series does not validate addresses and always returns a valid result.
- If you enable this option, the Vertex DM must be running when customers create accounts. If the connection to Vertex is offline, you can change this option to skip validating ZIP codes. This allows customers to create accounts.

To specify whether to validate ZIP codes:

1. Open the CM configuration file (*BRM\_home/syslcm/pin.conf*).
2. Change the **tax\_valid** entry:
  - To enable ZIP code validation, enter **3**.
  - To disable ZIP code validation, enter **0**. This is the default setting.
3. Save and close the file.

You don't need to restart the CM to enable this entry.

## 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. Open the CM configuration file (*BRM\_home/syslcm/pin.conf*).
2. Add the following entry:



```
- fm_rate tax_return_juris itemize
```

To summarize the taxes, change the entry to **summarize** instead of **itemize**.

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\_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

## 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.

### *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.